# Improving performance quality and user experience in the PULS News Mining system

Du, Mian

University of Helsinki, Department of Computer Science

2012

Date of acceptance Grade

Instructor

# Improving performance quality and user experience in the PULS News Mining system

Mian Du

Helsinki September 17, 2012

UNIVERSITY OF HELSINKI
Department of Computer Science

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

| Tiedekunta — Fakultet — Faculty | Laitos — Institution — Department |
|---|---|
| Faculty of Science | Department of Computer Science |

Tekijä — Författare — Author
Mian Du

Työn nimi — Arbetets titel — Title
Improving performance quality and user experience in the PULS News Mining system

Oppiaine — Läroämne — Subject
Computer Science

| Työn laji — Arbetets art — Level | Aika — Datum — Month and year | Sivumäärä — Sidoantal — Number of pages |
|---|---|---|
| | September 17, 2012 | 0 pages + 48 appendices |

Tiivistelmä — Referat — Abstract

Pattern-based Understanding and Learning System (PULS) can be considered as one key component of a large distributed news surveillance system. It is formed by the following three parts,

1. an Information Extraction (IE) system running on the back-end, which receives news articles as plain text in RSS feeds arriving continuously in real-time from several partner systems, processes and extracts information from these feeds and stores the information into the database;
2. a Web-based decision support (DS) system running on the front-end, which visualizes the information for decision-making and user evaluation;
3. both of them share the central database, which stores the structured information extracted by IE system and visualized by decision support system.

In the IE system, there is an increasing need to extend the capability of extracting information from only English articles in medical and business domain to be able to handle articles in other languages like French, Russian and in other domains. In the decision support system, several new ways of Information Visualization and user evaluation interfaces are required by users for getting better decision support.

In order to achieve these new features, a number of approaches including Information Extraction, machine learning, Information Visualization, evolutionary delivery model, requirements elicitation, modelling, database design approach and a variety of evaluation approaches have been investigated and adopted. Besides, various programming languages such as Lisp, Java, Python, JavaScript/Jquery, etc. have been used. More importantly, appropriate development process has been followed. This thesis reports on the whole process followed to achieve the required improvements made to PULS.

ACM Computing Classification System (CCS):
A.1 [Introductory and Survey],
I.7.m [Document and text processing]

Avainsanat — Nyckelord — Keywords
Information Extraction, Information Visualization, Machine Learning, Decision Support

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — övriga uppgifter — Additional information

# Contents

# 1 Introduction

Along with the development of web and web technologies, the amount of electronically accessible documents has been growing exponentially. Specifically to newspaper articles, a huge number of newspaper websites, which cover all domains globally, have been constructed providing online daily news. Some systems then try to gather related articles from a specific domain in order to serve the specific interests. In the business area, Esmerk [Esm11], for example, monitors thousands of newspapers, magazines, trade journals, web sources and press releases from 130 countries in 30 different languages for retrieving articles related to business information every day mostly by manual selection. Europe Media Monitor [EMM11] is another example which gathers over 40,000 reports every day from news portals world-wide in 43 languages, classifies the articles, analyzes the news texts by extracting information from them, aggregates the information, issues alerts and produces intuitive visual presentations of the information found. The service provided by EMM is known as document retrieval, which generally applies a keywords matching process to filter out irrelevant news articles that do not contain the combination of the keywords and return the possibly related ones. For the purpose of disease outbreak surveillance, the keywords could be any disease names or their synonyms. However, only keywords matching process performs problematically when handling natural-language text and the following two types of wrong results would be easily returned,

1. a large number of irrelevant documents that contain the disease name but describe something else rather than an outbreak event, such as medical research news discussing about some disease;

2. adding more keywords like "outbreak", "epidemics" or "pandemics" to the query in order to increase the precision, on the other hand, would eliminate many relevant articles which do not contain those specific keywords mentioned above since the number of different terms or sentences to describe an outbreak in plain text is almost infinite.

Moreover, these *article level events* returned are in fact not sufficient for the purpose of outbreak surveillance because the real information we need in these articles about what (disease), where (location), who (human or animal), when (time) and other related information like number and status of victims is still unknown. Aiming to find all disease outbreaks in a specific country reported in the newspaper articles

in a day, one might try to read these possibly relevant articles and acquire really valuable information from them, the manual work done to find them could often be tedious and inefficient. Considering the overwhelming number of newspaper articles every day, there is an increasing need for an automated method to process a large number of articles efficiently and precisely analyze them and extract the valuable information from them. Such method is generally known as Information Extraction.

Information Extraction was introduced in early 70s for extracting specific data from natural language text such as literature resources and newspaper articles. It has been widely used in many areas. There are a number of Information Extraction systems developed in recent years to support news surveillance system to better track disease outbreaks in medical domain, daily events in business area, etc. Pattern-base Understanding and Learning System (PULS), developed by University of Helsinki, provides such support for news surveillance. It is formed by the following three main components,

1. an Information Extraction (IE) system running on the back-end, which receives news articles as plain text in RSS feeds arriving continuously in real-time from several partner systems, processes and extracts information from these feeds and stores the information into the database;

2. a Web-based decision support (DS) system running on the front-end, which visualizes the information for decision-making and user evaluation;

3. both of them share the central database (DB), which stores the structured information extracted by IE system and visualized by decision support system.

Originally, PULS was focusing on Information Extraction from English newspaper articles in disease outbreak and business domains. Recently, the focus of PULS has been extended to be able to handle articles written in other languages and cross-border security domain in order to meet users' requirements. Besides, in the DS system, several new ways of Information Visualization and user evaluation interfaces are needed for better decision support. This thesis reports the whole work progress made to develop new functionalities for improving performance quality and user experience in the PULS News Mining system. It presents a detailed, in-depth account of a body of work that has been published in several research papers, including,

1. "Predicting the relevance of event extraction for the end user", Silja Huttunen, Arto Vihavainen, Mian Du, Roman Yangarber [HVD13]

2. "Techniques for multilingual security-related event extraction from online news", Martin Atkinson, Mian Du, Jakub Piskorski, Hristo Tanev, Roman Yangarber, Vanni Zavarella [ADP13]

3. "Building support tools for Russian-language information extraction", Mian Du, Peter von Etter, Mikhail Kopotev, Mikhail Novikov, Natalia Tarbeeva, Roman Yangarber [DVK11]

4. "Fast Adaptation of an Information Extraction System to the Russian language", Lidia Pivovarova, Mian Du and Roman Yangarber (submitted for review to COLING-2012, the 24th International Conference on Computational Linguistics)

The author of the thesis is a co-author on the above publications.

In this chapter, objectives and motivations of these improvements will first be discussed followed by the approaches used to develop the improvements. Finally, an overview of this project will be presented.

## 1.1 Project Objectives

This project involves the development of new features to improve performance quality and user experience in the PULS News Mining system. Specifically,

- In **PULS IE system**, besides improving the quality of extraction outcomes from English articles, a French pipeline and a Russian pipeline are required to be integrated into the system to handle articles in French and Russian. The cross-border security sub-system needs to be built in to extract security incidents from related newspaper articles.

- In **PULS DS system**, the current table view, which directly presents the information from database table, should be improved for better decision support (e.g., be able to sort by any field, provide more advanced query, be able to save queries, show more available fields, etc.). On current document page, a better way of verifying an event and assigning relevant level of an event are

required. More visualization interfaces including list view which groups similar events and display the list of groups, timeline view, graph view, map view, etc., would provide much better decision support for users.

## 1.2 Project Motivations

The improvement proposals stated above would benefit both PULS system developers and its users while providing more research opportunities for new topics.

### 1.2.1 Features to support both development and end-users experiences

1. Since the extraction results from the French and Russian pipelines are normalized into English, the French and Russian pipelines and the cross-border security sub-system would extend the capability of PULS to extract valuable information from more resources and hence enlarge the knowledge of PULS in the sense of both quantity and content and make PULS to be able to meet more clients' requirements.

2. The new document view provides better evaluation UI so that,

   - users can verify, delete and create an event more efficiently;
   - users can restore the whole document if earlier edits are not wanted anymore;
   - users can assign the relevance level to an event.

   PULS can then learn more from users' feedback and be able to improve the extraction in multiple ways.

3. New visualization interfaces for decision support system will better present the information extracted by PULS to users and users may find their required information and generate useful report more efficiently.

### 1.2.2 Benefits for research purposes

These improvements would potentially create more opportunities to improve PULS system in the future. For example, the new function for users to assign relevance to an event provides PULS system the opportunity to learn how to distinguish high relevance events and low relevance ones. By using these user input data and machine

learning algorithms, PULS may build classifiers to automatically assign relevance to new events and it provides opportunity to research on how to improve the classifiers.

In addition, since the quantity of information would increase after integrating the French and Russian pipeline into PULS, cross-document merging is another interesting topic that tries to group and verify same events from multiple document resources and languages in order to improve the accuracy of extraction and reduce duplication [Yan06].

## 1.3 Approaches

A number of approaches have been used to develop these new features including,

- **Information Extraction**: Information Extraction [GaW98] is the key technology applied in PULS IE system to acquire syntactic and semantic structure of the text, extract only specific kind of information from the unstructured text.
- **Machine learning**: Machine learning is used for building learning classifiers to improve PULS system in several ways.
- **Information Visualization**: Information Visualization is used in PULS DS system to create an intuitive way to convey abstract information in order to assist people to see, explore or understand a massive amount of information immediately [CMS99].
- **Evolutionary Delivery Model**: since the improvement requirements are not initially stable and new requirements always come up with research ideas and through continual interaction with end-users and project partners in the industry, it is best to use a model which builds on successive prototypes to improve PULS. One example of such model is called *Incremental Development Practices* (IDP) [McC96]. The phrase *IDP* refers to the development practices that allow a program to be developed and delivered in stages. IDP reduces risk by breaking the project into series of small sub-projects which are much easier to be completed than single monolithic project. Evolutionary delivery model which is one of the life-cycle models that support incremental development has been chosen. Figure 1 illustrates how the system works [McC96].

- **Requirements elicitation**: in order to successfully create the adequate and correct requirement specification, the approach of requirements elicitation is used to obtain enough requirements from a variety of sources including: PULS changes, literature sources, users' specific requirements, requirements from researchers' point of view.

- **Modelling**: the most common and powerful design approach called *modelling* is chosen for the functional designs. By using the UML which is one of the modelling languages, all functional requirements identified in the requirement stage have been easily transferred into actual functional designs one by one.

- **ER to the relational database mapping**: database of PULS are improved according to the database design approach proposed by Elmasri and Navathe [ElN03].

- **Evaluation approaches**: several comprehensive evaluation approaches are adopted to evaluate and test the new features in both functional and non-functional point of view. Unit testing and integrity testing have been performed for internal testing. Feature checklist and cooperative evaluation have been performed for external evaluation.

## 1.4 Overview

The thesis is organized as follows,

- **Chapter 2 - Background**: the development of improvements starts with the investigation of the background of this project. In this chapter, Information Extraction, text mining and Information Visualization will be presented first followed by the programming languages used to develop these improvements.

- **Chapter 3 - Requirements**: in this chapter, current PULS system will be introduced first. A number of requirements for improving PULS from different resources will then be elicited and analyzed. Finally, all functional and non-functional requirements gathered will be assigned priorities to clearly present which requirements are needed to be firstly considered.

- **Chapter 4 - Design**: in this chapter, new functionalities designed using UML will first be presented, followed by user interface design and database design.

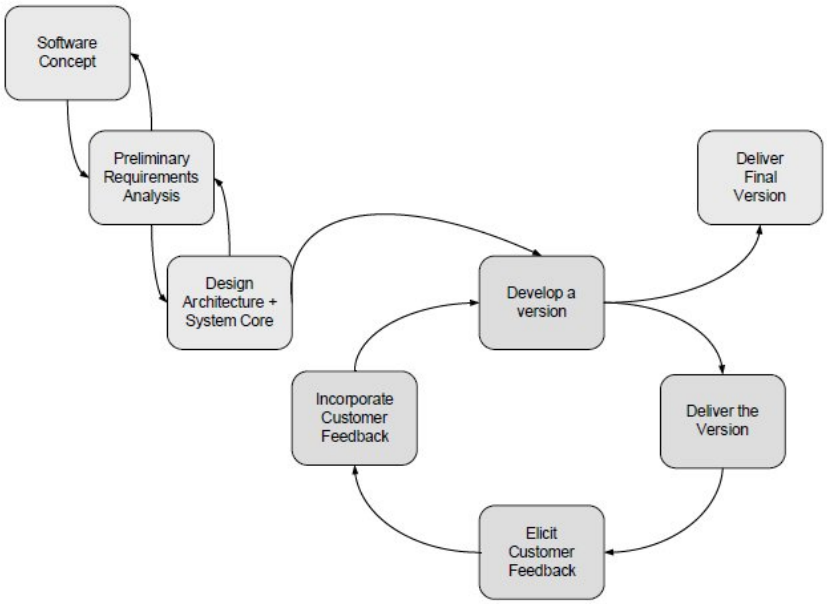


Figure 1: Evolutionary Delivery Model [McC96]

- **Chapter 5 - Implementation**: in this chapter, implementation technologies will first be introduced. All functionalities designed will be implemented and presented by first realizing the core functionalities and then implementing the optional functionalities. Finally, database implementation according to the database design presented in the design phase will be presented.

- **Chapter 6 - Evaluation & Testing**: in this chapter, internal testing which aims to ensure all the new functionalities can work properly will first be presented. External evaluation with the attempt to ensure these improvements can satisfy the actual users' needs will then be presented. Finally, evaluations of the main functionalities will be presented.

- **Chapter 7 - Conclusion & Further Work**: in this chapter, conclusion of this project including what have been done and what have been shown in the report will be provided first. Further work will be outlined.

# 2 Background

The development of this project starts with the investigation of the background related to PULS. PULS is one key component of a large distributed news surveillance system formed by an Information Extraction (IE) system and a Web-based decision support (DS) system. The IE system extracts structured information from unstructured news articles and stores information into the database while the Web-based DS system is used for visualizing and presenting the structured data to the end users. The concept and background of those terms: "Information Extraction" and "Information Visualization" are therefore needed to be well understood. Besides, technologies that will be used to develop this project should also be investigated. In this chapter, background of Information Extraction and common structure of IE system will first be presented in Section 2.1. The principle of Information Visualization and how it is used in decision support system will then be presented in Section 2.2. Background of programming languages used in this project will be presented in Section 2.3.

## 2.1 Information Extraction

Beyond document retrieval, Information Extraction (IE) is the process to automatically extract pre-specified information from natural language text [GaW98]. Information Extraction applies a number of formal linguistic analysis methods to acquire syntactic and semantic structure of the text, extract only specific kind of information from the structured text and finally store the information into a database for later query. Any document that does not contain this specific kind of information is considered irrelevant and will be discarded.

For instance, one might try to scan general newswire texts for infectious disease outbreak event containing *how many* victims are suffering from *which disease* in *where*. A newswire text containing sentence *"In Benin cholera has killed five people yesterday in a rare dry-season outbreak."* would then produce the following template output[1] (see Table 1) through correct IE process. Any information other than disease outbreaks in the text is considered as irrelevant and will be discarded. By processing a large number of newswire texts every day, this type of structured template outputs of information are then stored in the database in order to build the knowledge base

[1]Term *yesterday* in the text is resolved to 01.02.2010 since the reporting date of the news is 02.02.2010.

of disease outbreaks for the purpose of outbreak surveillance.

| | |
|---|---|
| **Disease** | Cholera |
| **Country** | Benin |
| **Time** | 01.02.2010 |
| **Total** | 5 |
| **Victims** | people |
| **Status** | dead |

Table 1: a template output example produced by IE process

Information Extraction should not be confused with technologies which might appear to be similar to IE such as information retrieval (also called *document retrieval*) and text categorization.

Information retrieval (IR) is a more mature technology often used before IE process to retrieve relevant documents from large dataset. IE then extracts specific events from these relevant documents. IR and IE are therefore complementary and it is potentially more powerful and efficient to combine these two technologies in text processing.

Similar to IR, Text categorization (TC), which is used to classify news stories into different categories, is also a key technique for organizing large text dataset before IE process. While IR focuses on retrieving documents containing one specific type of information such as disease outbreak and widely ignores other information that could be also important, TC tries to classify documents containing all kinds of important information into their own categories such as medical, business, security, etc., for later information needs. IR normally applies keyword matching to filter out irrelevant documents while TC is mainly based on statistical analysis of word frequency by text classifiers [Joa02].

By using a combination of these techniques stated above, the future Internet search engines could retrieve specific detailed information like the one stated in Table 1 from large text collections and organize information for intelligent, semantically normalized search.

A detailed research on development history of IE systems is presented in Appendix 1a.

### 2.1.1 Common structure of IE systems

IE systems, depending on their purpose, may have different structures which could be difficult to classify. It is however possible for newcomers of IE field to grasp some typical process stages involved in IE process by looking at J.Hobbs's description of the *Generic Information Extraction System*. According to Hobbs, a generic IE system is a *"cascade of transducers or modules that at each step add structure and often lose information, hopefully irrelevant, by applying rules that are acquired manually and/or automatically"* [Hob93]. He concluded in the paper that, despite the particular combination of modules characterizing an IE system, generally most systems would perform the functions of some or all modules as shown below,

1. **Text Zoner**: divides the text into text segments (heading, body, etc.).
2. **Preprocessor**: converts the text segments into a sequence of sentences, and for each sentence, converts each word the sentence contains into a lexical item with its lexical attributes (e.g. part of speech tag, basic form, etc.).
3. **Filter**: Filters out irrelevant sentences from the sequence of sentences.
4. **Preparser**: identifies and groups a sequence of lexical items into small-scale structures if possible (e.g. noun phrases, verb groups, apposition, etc.).
5. **Parser**: produces a set of parse tree fragments, which describes the structure of the sentences by analyzing the relationships of those lexical items and small-scale structures.
6. **Fragment Combiner**: tries to merge a set of parse tree or logical form fragments into a whole parse tree or logical form for the whole sentence.
7. **Semantic Interpreter**: generates the semantic structure or meaning representation from the whole parse tree or parse tree fragments.
8. **Lexical Disambiguation**: turns the ambiguous predicates in the semantic structure into specific and unambiguous predicates.
9. **Co-reference Resolution or Discourse Processing**: identifies and links the same entity with same or different descriptions in different parts of the text, and builds a network-like semantic structure of the whole text.

10. **Template Generator**: takes the semantic structure produced by the above natural language processing modules and produces the templates in the required form.

In general, an IE system has processing modules that perform the above tasks. However, not all systems exhibit all of these modules separated like this. For example, *Text Zoner*, *Preprocessor* and *Preparser* are usually merged into one component which takes the text as input and outputs the separated sentences containing lexical items or small-scale structures. The sequence of these modules could also be different for different systems (in particular, *Fragment Combiner* and *Lexical Disambiguation* may happen in the reverse order).

### 2.1.2 Methods of extraction

The resulting output consists of required items as slot values of a structured template (Table 1). Based on the linguistic analysis output produced by components 1 to 6, extraction patterns are usually used in *Semantic Interpreter* to match facts. These facts are then used to fill the slots of the result template. An extraction pattern is a text pattern which contains a space for a specific token and its surrounding context. While the surrounding context is fixed, the token is variable. For instance, $X$ **was/were infected by** $Y$ **on** $Z$ is a sample pattern. It could be used for matching outbreak event from a sentence like "18 people were infected by H1N1 on Friday." **... were infected by ... on ...** is the fixed surrounding context in this example;$X$, $Y$ and $Z$ are the variable tokens. According to the definition of required slots by the IE system, $X$ here could be any human noun group (e.g. *18 people*, *a 38-year-old Brazilian woman*, etc.), $Y$ is any infectious disease name and $Z$ may be any representation of date group (e.g. *Friday*, *24th, May*, etc.).

An IE system usually has a number of such extraction patterns to match required facts. Different systems, according to their purposes and domains, may have totally different patterns. Finding extraction patterns is therefore considered as the core task of IE system building since the quality of the resulting template largely depends on the quality of these patterns. In general, there are two following different ways of obtaining the suitable patterns.

- **Knowledge Engineering Approach**, which manually defines the extraction patterns by linguists and knowledge experts of the required domain [AHB93, MTU01, CMB02, JKR06, SDN07].

  - **Advantages**: the precision of facts extracted by these manually defined patterns is normally high.
  - **Disadvantages**: this approach requires a lot of time to be spent on creating and evaluating the suitable patterns; it is usually domain and language specific; the recall of the facts could be low since it is difficult to come up with all possible patterns that can match all kinds of natural language descriptions of the fact especially manually.

- **Machine Learning Approach**, by using machine learning algorithms, this method automatically identifies the essential regularities for Information Extraction from a training set that consists of suitable annotated texts, and these essential regularities are then used for creating extraction patterns. Example IE systems using this approach includes *AutoSlog-TS* [Ril93, Ril96, PaR06], *CRYSTAL* [SFA95, Sod96], *PALKA* [KiM95], etc.
  - **Advantages**: the recall of facts extracted by using these automatically acquired essential regularities is normally high especially when the training set is good enough in terms of size, regularity, domain specificity, etc.; it is usually much faster than manual approach to obtain a large set of useful patterns; theoretically it is domain and knowledge independent.
  - **Disadvantages**: while obtaining the useful patterns quickly, this approach also brings a number of irrelevant patterns that could dramatically decrease the precision of facts; finding good training and testing sets of text is also a challenging task since bad training and testing set results in disappointing patterns.

Both of these approaches continue to be used in parallel for different systems depends on the nature of the task and the amount of noise in the unstructured data.

Besides pattern-based extraction, statistical methods have been introduced in recent years since the extraction patterns were found to be too brittle for more noisy unstructured sources. Two kinds of techniques were deployed in parallel: *generative models based on Hidden Markov Models(HMM)* [BMS97, SMR99, BDS01, AgG04] and *conditional models based on maximum entropy* [BSG98, Rat99, MFP00, KlM02, Mal02]. Although the statistical methods for exaction is newer, there is no clear winner until now. The characteristics of two methods can be summarized as follows,

- **Pattern-based methods**

  - driven by hard predicates.
  - easier to develop.
  - more useful in specific domains where human involvement is both essential and available.

- **Statistical methods**
  - decision made by weighted sum of predicate firings.
  - more robust to noisy data.
  - more suitable for open-ended domains such as opinion extraction from Blogs.

This paper is focusing on pattern-based extraction method since PULS system extracts facts from several closed domains. In Chapter 3, we will present the detailed structure and extraction method of PULS.

## 2.2 Information Visualization and Decision Support System

Now, we have the *facts* which are extracted by the Information Extraction technologies described in the previous section. These facts are normally stored in the database. How to use these facts effectively brings us the terms "Information Visualization (IV)", "Data Mining (DM)" and "Decision Support (DS)". In this section, principles of these terms and how these principles are taken into account when creating a decision support system are introduced and investigated.

### 2.2.1 Information Visualization

IE technologies have enabled the automatic collection of huge amount of information from a variety of resources in a short time. Rapid growing data also imposes a challenging task for presenting them appropriately and efficiently. Data summarized and presented in a suitable and illustrative manner would demonstrate and reveal valuable and significant facts to users, while an inadequate representation confuses users and dramatically decreases their enthusiasm to explore potentially very useful information. A visualization format is normally the best choice for this task.

Information Visualization includes all developments and progresses made in *Data Visualization*, *Infographic*, *Scientific Visualization* and *Visualization Design*. Having

been organized appropriately, everything can be considered as a type of information: tables, graphs, maps, and even the plain text, whether they are static or dynamic, will provide us some means to explore insight into where we can actually find out the meaning and all kinds of relationships, and possibly understand the knowledge that can not be easily identified by any other means. Now, in scientific and technological research, the term *Information Visualization* is generally applied to visualization of large-scale and non-numeric information resources.

Information Visualization is committed to create an intuitive way to convey abstract information in order to assist people in *using their vision to think* [CMS99]. By utilizing the advantage of human eye's broad bandwidth pathway into the mind, all kinds of visual representations and interaction techniques enable users to see, explore or understand a massive amount of information immediately.

History of IV is presented in Appendix 1b.

**Application areas of IV** Among the Information Visualization process, visual data to be collected is not the result of some mathematical models or large data sets, but the abstract data with its own inherent structure. Examples of such data include:

- WWW site content;
- the operating system file space;
- data returned from a variety of database query, such as digital libraries;
- compiler or other program's internal data structures, or trace information of large-scale parallel program.

Not only the appropriate data needs to be collected and processed, the type of graphical elements selected to display these different types of data is also very important. There are a number of different types of graphical elements that we can choose from to display data, such as bar chart, line chart, sparklines and bullet charts [Tuf01, Few06].

By selecting different types of graphical elements and interactive methods according to the data, IV has increasingly become the key element in a variety of different areas such as: scientific and technical research, digital library, data mining, financial data analysis and market research, manufacturing process control, Crime map, etc.

In current information world, IV plays a very important role aiming to answer the following questions.

- How to organize the information explosion?
- How to understand the information overload?
- How to find relationships among nested information?
- How to show various forms of data to help people explore the information?

Now, let's take a look at some examples.

- **Where does my money go?** Before Christmas 2009, the British government announced that it would open local spending reports, and wheredoesmymoneygo.org was born [WDM11]. Users can view all the British government expenditure in all domains and areas during the past 6 years. Through operating this project, the government hopes to collect more detailed information, including local consumption. Since all graphics are rendered with the corresponding size of amount, and different colors for different domain or areas, the proportion of each part's expenditure can be viewed clearly and intuitively (Figure 23 in Appendix 1c). Users may also go deeper to check more detailed spending for a specific domain or area by interacting with the graphical user interface.
- **5 Years Infosthetics** *"Assuming we have a total of 2000 data, each screen display 50, how many screens are needed?"* [5yrs11] IV gives us an surprising answer. To celebrate Infosthetics Forum's 5th anniversary, the designer put a total of nearly 2000 items on the same page (Figure 24 in Appendix 1c) by using various IV methods [5yrs11]. Each piece in the left side represents a project. The different colored boxes in the piece represent different categories that a project belongs to. The legend of the categories and other properties of these projects are placed in the right side. Users may click any piece for viewing the detailed information of that project including project name, description, link, highlighted categories and other properties in the legend; or they can also perform multi-dimensional query by simply selecting one or more categories and other properties in the legend. Just several clicks would help users to acquire the required information effectively.

In short, IV provides access to extensive and profound knowledge. Through investigating and analyzing the characteristics of the data, an appropriate design of visualization using IV techniques would dramatically demonstrate more potential value of abstract data. We will come back to see how IV is applied in DS system and PULS in the following section and the rest of the paper.

### 2.2.2 Decision Support System

Decision support systems (DSS) is a computer-based information system which assists users to make decisions by utilizing the data, models, knowledge and human-computer interactions provided by the system [SpC82]. It is a more advanced outcome generated by the development of Management Information System (MIS). Aiming at improving the quality of decision making, DSS provides an interactive environment and useful tools to decision-makers for compiling useful information from various information resources, analyzing the problem, modeling and simulating the decision-making processes towards the final solutions.

The development history of DSS is presented in Appendix 1d.

**Structure of DSS** The Dialog-Data-Modeling (DDM) architecture proposed by Sprague and Carlson is accepted by most academics as the initial structure of DSS [Mar99, Pow02]. They describe that DSS has three fundamental components [SpC82]: the *data module*, the *model module* and the *user interface module*. When DSS was combined with ES, the *inference module* was also added into DSS:

- **data module** contains the database and its Database Management System (DMS) [TuA97]. DSS database contains a large number of internal information (such as internal accounting data), or external data (such as financial indices). These raw data would need to be gathered and extracted to the data format suitable for decision-makers to manage, analyze, update and retrieve [SpC82].
- **model module** includes the Model Base (MB) and its management system (MBMS). MBMS integrates various decision-making models to analyze the internal and external information from the database. An example model could be the mathematical model which analyzes and simulates a complex problem, comes up with feasible solutions and help the user to choose a solution from those options. MBMS also includes the modeling language for users to cus-

tomize the models or build their own models [Gac04].The basic abilities of MBMS include [HCM04]:

1. satisfying the users' needs of different models.
2. capability of integrating model and data.
3. providing easy-to-use interface.
4. capability of sharing models.

- **inference module** is formed by the KB, Knowledge Base Management System (KBMS) and the inference engine component. In order to solve many unstructured or semi-structured problems, specialized knowledge is required besides the standard features of DSS. So in the modern DSS, the KBMS is also an important sub-system in addition to DBMS, MBMS and DGMS [Gac04].

- **user interface module** is the interactive part of the DSS. It accepts and inspects the user requests, calls the functional components within the system to invoke the model runs, data analysis and knowledge inferences to effectively solve the decision problem [TAL08]. *User interface module* has three main actors: the user, computer hardware and software systems. The communications between human and DSS can be divided into three parts [Mar99]:

  1. **The Action Language**: refers to any way the user would use to communicate with the DSS, such as keyboard, mouse and any other controlling hardware of software instructions.
  2. **Display or Presentation Language**: refers to the DSS output of information in any form that the user can explore, such as monitor, printer, etc.
  3. **Knowledge Base**: contains any required knowledge that the user needs to know to use the DSS effectively, such as user manuals.

The advantages of the graphical format for displaying information, as described in the previous section, make Information Visualization best serve the displaying and presenting purpose of DSS output in the user interface module. By integrating IV into DSS, common presenting problems like "information overload" caused by the large quantity of data from database, model results and knowledge base are easily resolved. The more advanced interactive IV enables user to navigate the decision support result more effectively and also provides a much easier tool for decision

makers to communicate with DSS. Almost all DS systems integrate interactive IV in its user interface module. The most commonly used graphical displays are *line chart*, *bar chart* and *pie chart*. More advanced graphical tools such as *dashboard*, *relational graph* or combined graphical formats are also popular. By selecting suitable formats from these options according to the DSS output, DSS simplifies the profound scientific questions into a visual image which gives full play to the human cognitive ability. Not only does DSS with IV facilitate the researchers to study and analysis the problems, but it also provides a powerful tool for them to communicate with it to acquire more powerful decision support. In the following, we will see a real example of how IV is applied into DSS in the PULS system.

## 2.3 Programming languages

Various technologies including XML, Lisp, JAVA, Python, JavaScript/Jquery, machine learning algorithms, etc., have been investigated and used in PULS system. How these technologies are used in PULS system will be introduced in Chapter 5. An introduction of XML is presented in Appendix 1e. This section briefly describes the background of LISP which is the main programming language of PULS.

### 2.3.1 Lisp

Lisp is the second oldest high-level programming language in use today and the name *Lisp* derives from *LISt Processing* [McC58]. Since 1960, when J.McCathy first published Lisp at MIT in the United States [McC60], it had been quickly accepted by researchers as the most popular programming language in the Artificial Intelligence (AI) domain and various achievements of AI are contributed by Lisp. While it armed a generation of AI scientists as the powerful weapon to endow the machine with artificial intelligence, Lisp is also widely used as an expressive language for stating algorithms in computational linguistics [GaM89].

**Why Lisp in NLP** In NLP, we often manipulate symbols (words, phonemes, parts of speech) and structures objects (sequences, trees, graphs) which are made from the basic symbols. The diversified structure of natural language text could be very difficult to represent using normal data structures in programming languages. For example, a common JAVA array with same class of instances is not suitable to represent a sentence containing the combination of different types of symbols

and combined structural objects. Lisp is however a high-level language that we can use to directly operate on those symbols and structures without worrying about how these high-level concepts are represented in the machine. Besides lists, Lisp provides many other kinds of complex data structures like hash tables, vectors that place no limitation on the data type of their elements. This makes it extremely easy for Lisp programmers to create any kind of desired objects according to the different purposes and contents in NLP.

Also, the concept of *recursion* is a fundamental characteristic of NLP. Linguistic objects are described by recursive data structures and operations on these structures are naturally expressed as recursive algorithms [GaM89]. Lisp, unlike some other programming languages, is natural for expressing such algorithms by placing no restrictions on procedures calling themselves directly or indirectly.

A short introduction of "what made Lisp different?" can be found in Appendix 1f.

## 2.4 Chapter summary

In this chapter, we have started with investigating the background of Information Extraction and common structure of IE system. Then, the principle of Information Visualization and how it is used in decision support system have been introduced. Finally, the main techniques used in this project have been presented.

# 3 Requirements

The analysis of customer requirements is one of the basic foundations of establishing the market strategy of an enterprise. Similarly, adequate requirements are vital to develop an appropriate software project. The degree of understanding of user requirements may determine whether a project will be successful or not. In order to ensure that PULS will be improved in the right direction where user requirements can be satisfied, we have elicited requirements from various sources and have been communicating with users throughout the project development process. In this chapter, the elicitation of appropriate requirements from different sources will be presented in Section 3.1. These requirements will then be analyzed and classified into functional and non-functional requirements in Section 3.2. After these two sections, we have a well-formed understanding of the users requirements and finally, these classified requirements will be assigned priorities in Section 3.3 to clearly present which requirements are crucial and need to be considered first.

## 3.1 Requirements Elicitation

Requirements elicitation is a process to confirm and understand users' needs. In general, requirements elicitation can be considered as the most difficult and important stage in developing any software project since any mistake or misapprehension of requirements elicitation which are later discovered will result in a lot of redoing. Steve McConnell has mentioned in his book called *Software Project Survival Guide*: *"If comparing the cost due to error correction in the early stage with the cost of error correction in the late stage, we may find that the latter one may be 50 - 200 times more than the first one"* [McC97]. In order to achieve requirements elicitation, a number of approaches may be used. In this section, we present 4 approaches used to elicit requirements in details.

### 3.1.1 PULS changes

PULS is formed by an IE system, a simple Web-based user interface and a shared database. The IE system, on the back-end, receives disease outbreak news articles from automated IR processes performed by Promedmail [Pro11] and JRC arriving continuously in real-time, and fetches business news articles that are manually picked up by Esmerk once a day.

Now, we have two more source partners providing more live business and cross-border security news articles. This requires us to extend our IE system to be able to handle a new domain - security and adapt our IE process for business to accommodate the new business source. Since both new sources collect news articles using automated IR process, PULS IE system also needs to be improved to deal with more noisy data.

In addition, majority of the news articles before was English and therefore the focus has been given to develop Information Extraction process for English only. Now, we receive also a large number of news articles in other languages like French, Russian, Spanish, etc. The processing pipelines for other languages generally need to be integrated in PULS IE system to process articles in these languages. This would extend the capability of PULS to extract valuable information from more resources and allow PULS to cross-validate the news events among articles in different languages. These changes have shaped the main requirements of improving PULS IE system as shown in Table 15 in Appendix 2.

The user interface which contains a table view and document view was designed for displaying the extraction results(see Figure 13 and Figure 14). The table view is formed by rows and each of them presents an extracted event and contains the event's slots values (e.g. country, disease, time, etc.) displayed as columns. From the table view, you may click a row to view the event in document view which contains the article text where the event is extracted from and more detailed slot information. The table view and document view was designed only for displaying English articles and events in disease outbreak domain. The IE system changes stated above hence requires the adaptation of user interface as shown in Table 16 in Appendix 2.

Accordingly, these IE and UI changes also result in the adaptations of database as shown in Table 17 in Appendix 2.

### 3.1.2 Literature resources

Users of PULS system can be classified into two groups, i.e., the users of extraction results from the PULS IE system and the users of PULS Web-based user interface. The first group of users includes the partner systems which receive the extraction results as XML files from PULS after the PULS IE process and the second group of users navigates the PULS user interface to view events of interest and use it as a

surveillance tool for disease outbreak, business activities, etc. While the first group is only concerned with the quality of the IE results, the second group is concerned with both IE results and the user interface. How to improve the quality of IE system results and how to better present the results to the end users are therefore considered as the two general objectives of PULS.

As an Information Extraction system, the main task of PULS is to maximize the quality of extraction results, similarly to the general objective of other IE systems. The term *quality* here would vary according to the extraction purpose and the evaluation metric. As stated in the previous chapter, since most of the sophisticated IE systems use the F-measure metric [Har11] which evaluates the results in terms of recall and precision to measure the quality and try to maximize its score [Rij79, DAR95, MaP98, ACE11], PULS also uses F-measure as the main evaluation metric. This general principle applied to PULS IE system introduces the requirements for improvements as shown in Table 18 in Appendix 2.

Similarly, as a Web-based application, the PULS Web-based user interface should try to apply the general guidelines for developing a good Web-based application as far as possible and consider them as the requirements for this project. In order to collect guidelines, I have investigated a number of literature resources. Some interesting ones include a book called *website management excellence* by Brigman [Bri96] and useful websites like *WebsiteTips.com* [WeT11], *WebReference.com* [WeR11], *useit.com* [use11]. From these resources, I have collected multiple guidelines for good Web-based application design as shown in the Table 19 in Appendix 2.

### 3.1.3 Users' specific requirements

As stated above, PULS system has two groups of users. During development of PULS, we always keep communicating with them to make sure that we understand the appropriate requirements from them. Through communicating with the users of PULS IE system, we have agreed on the desired structure and content of the IE results as well as the direction of improving PULS IE system. These specific requirements for PULS IE system are shown in Table 20 in Appendix 2.

Through communicating with users who use PULS user interface regularly to view their events of interest, we have agreed that the UI should provide more informative and dynamic views besides the simple table view (Figure 13) and static document view (Figure 14). Based on the investigation of Information Visualization and Deci-

sion Support System as described in the previous chapter, we planned to extend our current user interface to an interactive DS system which visualizes the extraction results using interactive IV techniques and has at least the features described in Table 21 in Appendix 2 for providing better decision support.

These changes required by users not only can provide much better visualization tools for them to absorb the information more quickly and use it to make decisions, they also provide a way for users to communicate and interact with the PULS system, through which the valuable verification and evaluation provided by user is recorded as expert knowledge which would be later used to improve the PULS IE system.

These changes in PULS IE system and DS system also require the changes of database accordingly as described in Table 22 in Appendix 2.

### 3.1.4 Improving requirements from researchers' point of view

Elicitation as stated above have formed the major requirements of improving PULS. During eliciting and developing the requirements, we have also come up with interesting and useful thoughts to improve both PULS IE system and DS system.

On the IE side, besides using traditional evaluation metric F-measure that judges the quality of the IE system results in terms of recall and precision, we propose a new metric for news surveillance system to help users to find their required events more quickly. The utility or relevance of an event, as described in [HVE11, HVD13], decided automatically by PULS IE system, is very useful for providing users better decision support. For example, an old disease outbreak event one year ago is absolutely correct event to be extracted if we use F-measure. Although such event could be important for background information, it should not attract people's attention as an outbreak event if it appears in some recent news articles. By giving such old events a lower relevance score and assigning recent outbreak events a higher score, people can more easily focus on the surveillance of current outbreaks by simply filtering out events with a lower relevance.

In order to make PULS system be capable of making such decisions, we use supervised machine learning algorithms to build a relevance classifier. The new edit feature provided on document view in PULS DS system described in previous section can be used by both users and developers to rate an event according to the relevance classification criteria (Table 2) and allows PULS to collect training/testing data required to build such relevance classifiers.

| **Criteria** | **Score** |
|---|---|
| New information; highly relevant | 5 |
| Important updates; on-going developments | 4 |
| Review of current events; hypothetical, predictions | 3 |
| Historical/non-current, background information | 2 |
| Non-specific, *non-factive* events; secondary topics | 1 |
| Unrelated to target domain; useless | 0 |

Table 2: Guidelines for relevance scores

Besides relevance classifier, we plan also to create a knowledge base to accumulate valuable knowledge other than extracted events. The knowledge base will be very useful for improving both PULS IE system and providing more decision support for PULS DSS users later. For example, Esmerk, one of our news feeding partners, provides sector tags they manually assigned for each business news article together with the article text. If we accumulate all of the sector tags first in PULS knowledge base, we would be able to use such knowledge to automatically classify the business sector for any new article in PULS IE system later using similar algorithms to those used for relevance classification.

These thoughts of improvements have generated requirements for PULS IE system in Table 23 and PULS DS system in Table 24 in Appendix 2.

## 3.2 Requirements Specification

In previous section, a number of raw requirements for improving PULS have been elicited from different sources; these requirements are merged and classified into functional and non-functional requirements in this section.

### 3.2.1 Functional requirements

Generally, functional requirements describe what the system is to do in response to user commands. Thus it is the functionality or services that the system is expected to provide to users. Since the functional requirements are mainly concerned with user requirements, it is beneficial to clarify the users of PULS system first. As mentioned before, users of PULS IE system are from PULS's partner organizations

in three domains requiring the IE extraction results while the users of PULS DS system are expecting to receive decision support based on the knowledge PULS has accumulated by its IE system. They form the two main user groups of PULS system and the detailed user groups can be found in Figure 2.

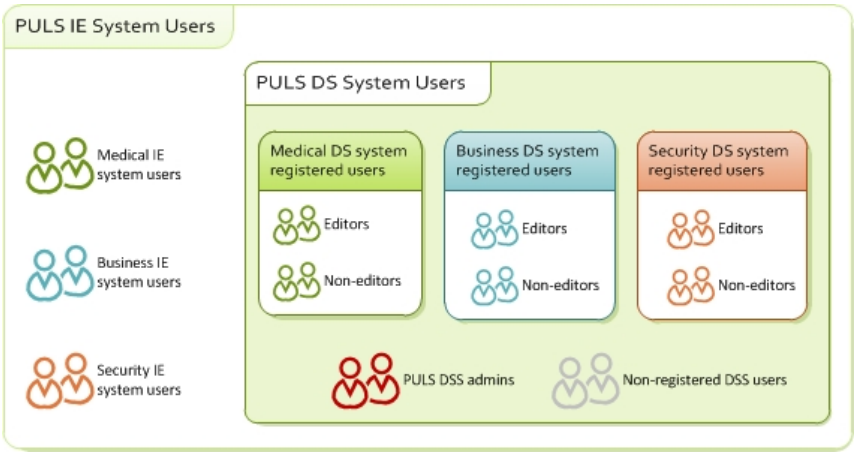


Figure 2: PULS user groups

As different users have different privileges and requirements, PULS system provides them with different functionalities as follow:

**Functionalities provided to PULS IE system users** PULS IE system users from all three domains have a common interest, i.e., to get good extraction results. Besides, each domain has their own specific expectations of features and functionalities. In order to meet these requirements, functionalities shown in Table 25 in Appendix 2 should be provided.

**Functionalities provided to PULS DSS users** All PULS DSS users are of course also implicitly PULS IE system users. Besides, for PULS DSS users, functionalities which visualize the extraction results using interactive IV techniques or enable users to communicate with PULS can be found in Table 26 in Appendix 2.

### 3.2.2 Non-functional requirements

Besides those functionalities provided to PULS users concluded in the previous section, other requirements we elicited before are basically non-functional requirements as shown in Table 27 in Appendix 2.

## 3.3 Requirements priority

In previous section, all the requirements elicited have been analyzed and classified into functional and non-functional requirements. In this section, those functional and non-functional requirements will be assigned priorities. In general, most projects can not achieve all the functionalities proposed in the requirement stage within a given time and resources. This principle applies to this project as well. It is therefore valuable to decide which requirements are essential and important in order to ensure that the core functionalities can be achieved first. Through discussions with project leader and users, we have decided to follow the priorities of functional requirements assigned in the Table 28 in Appendix 2 and the priorities of non-functional requirements defined in Table 29 in Appendix 2.

All the requirements with high or medium priority in these two tables (core requirements) need to be satisfied while those requirements with low priority will be considered if there is still time available (optional or advanced requirements). From Table 28, we can easily see that all the functionalities required by users have high or medium priority which will be firstly considered and developed. Functional requirements with low priority are mostly those features elicited for research purposes which require a large amount of experiments and time to achieve. non-functional requirements are normally with lower priority. Exceptional cases include those non-functional requirements which are very important and helpful to support the Functional requirements.

## 3.4 Chapter summary

In this chapter, a number of requirements have been elicited from different resources in Section 3.1. Those requirements have then been analyzed and classified into functional and non-functional requirements in Section 3.2. After these two sections, we had an overall understanding of user requirements. Finally, the priorities of functional and non-functional requirements have been decided in Section 3.3. Those priorities has clearly shown which requirements are core requirements that can be realistically delivered within the lifetime of the project and which requirements will only be considered as optional requirements afterwards. In the next chapter, designs of developing these requirements will be presented.

# 4 Design

The main purpose of design phase is trying to establish the architecture which defines the components, their interfaces and behaviors. In the previous chapter, specific requirements for improving PULS are collected and classified into functional and non-functional requirements. During design phase, these requirements are translated into system architecture and other actual plans which describe how to implement the system to meet these requirements including system architecture, algorithms, data structures, interfaces, etc., should be established. In this chapter, we will then focus on the architecture and functionalities of PULS designed using UML which are aiming to satisfy Functional requirements in Section 4.1. User interface design which mainly tries to meet non-functional requirements will then be presented in Section 4.2 Data structures will be introduced in Section 4.3.

## 4.1 Functional Design

Functional design is trying to establish the plan to accommodate the Functional requirements in the proper way. In this section, Functional requirements obtained from the requirement stage will be translated into actual design using UML.

### 4.1.1 UML Introduction

The difference between building an edifice and a kennel is obvious: we do not need to think about design for building a kennel. However, in order to develop an expected system which can accommodate user requirements, we need to concern with design. The best approach of design is modelling, to represent reality in a simple way. Modelling can:

- Help us visualize the system according to our required patterns.
- Allow us to specify the structure and behavior of the system.
- Provide us templates to construct the system much easier.
- Archive our decision.

Basically, modelling tries to break a complex system into sub-systems and separately design every sub-system. UML stands for Unified Modelling Language. It is

an object-oriented modelling language. As a modelling language, it enables designers to break this system into several sub-systems; it provides notations to describe this system in a clear way. Therefore, UML will be used to analyze and design this application. In this section, **Static Structure Diagram** which describes the structure of PULS system and its sub-systems will be introduced first in Section 4.1.2; **Use Case Diagram** used to show the functional view of this application will be presented in Section 4.1.3 followed by the **Class Diagram** which is used to show the static view of PULS system in Section 4.1.4; finally, **Activity Diagram** which demonstrates the dynamic view of PULS will be shown in Section 4.1.5.

### 4.1.2 Static Structure Diagram

Static Structure Diagram as shown in Figure 3 describes the static overall structure of required PULS system. From it, we can easily see that the required PULS system is formed by three PULS IE sub-systems, a Data component and three PULS DS systems for three domains. The existing components are marked by the red border and these components are required to be improved. Components with normal solid border are going to be developed to extend the capability of PULS system. Those components with grey dashed border are optional components and will be considered last.

### 4.1.3 Use Case Diagram

Functional requirements obtained in the previous chapter are considered as actual functionalities of PULS. The Use Case Diagram best illustrates the functionalities provided to different users.

**Functionalities provided to PULS IE system users** According to Table 25 and Table 28, PULS will provide PULS IE system users functionalities as shown in Figure 4.

**Functionalities provided to PULS DS system users** According to Table 26 and Table 28, PULS will provide PULS DS system users functionalities as shown in Figure 5.

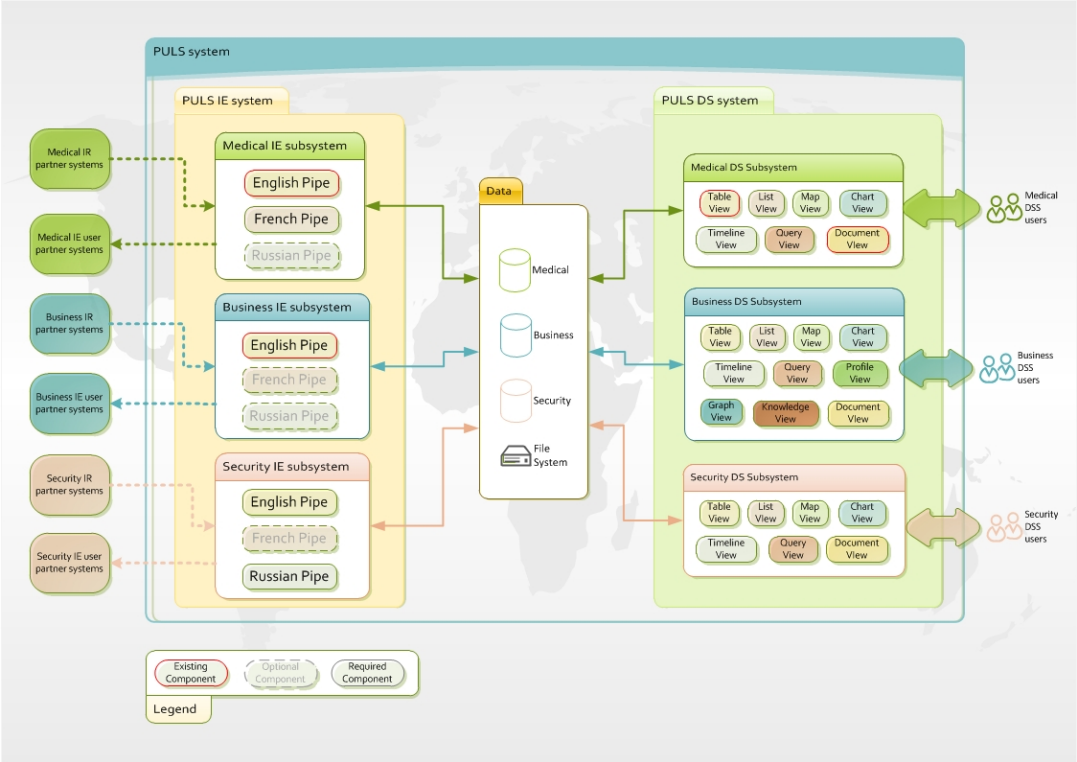


Figure 3: PULS Static Structure Diagram

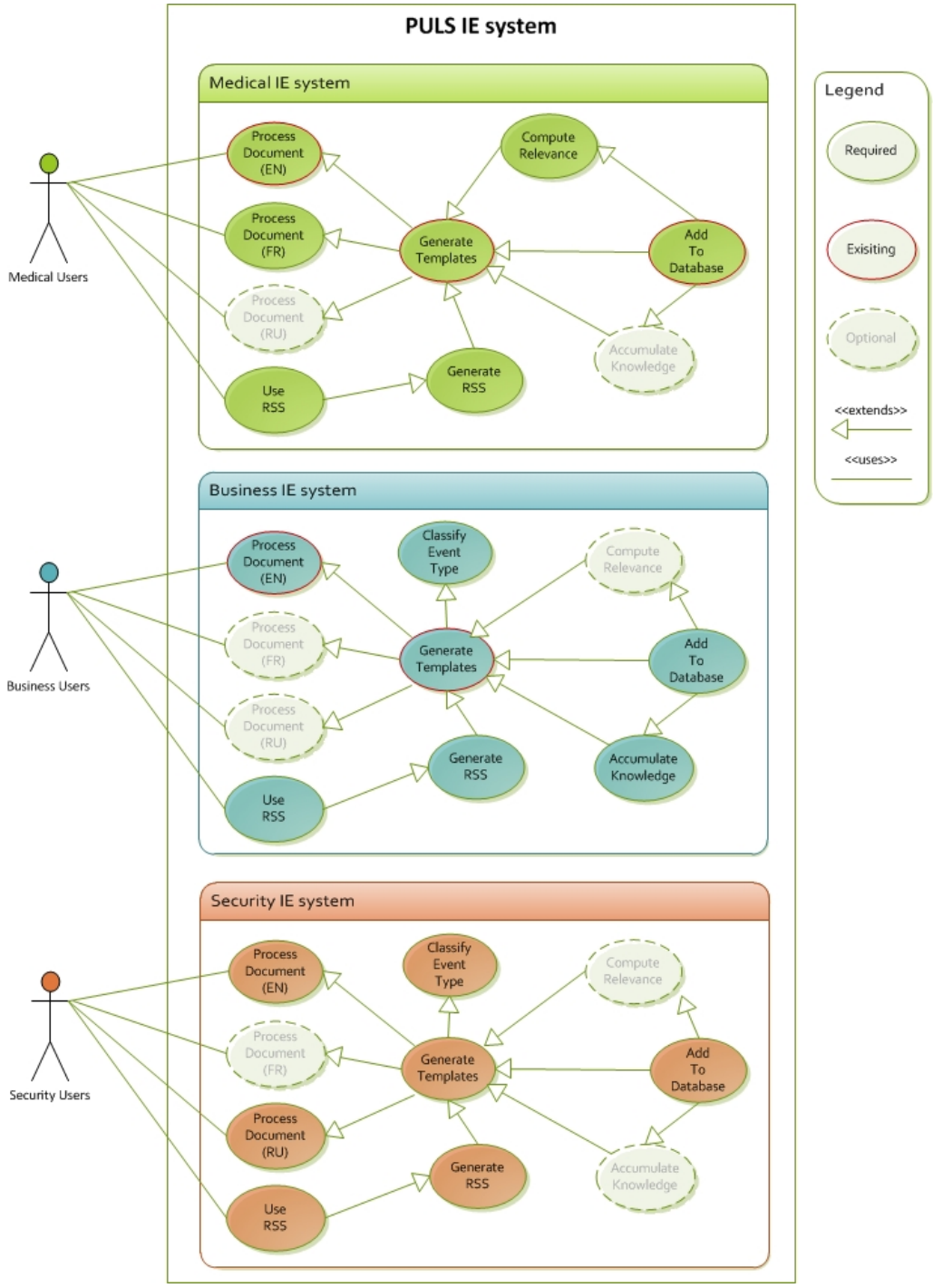


Figure 4: Use Case Diagram for PULS IE system users

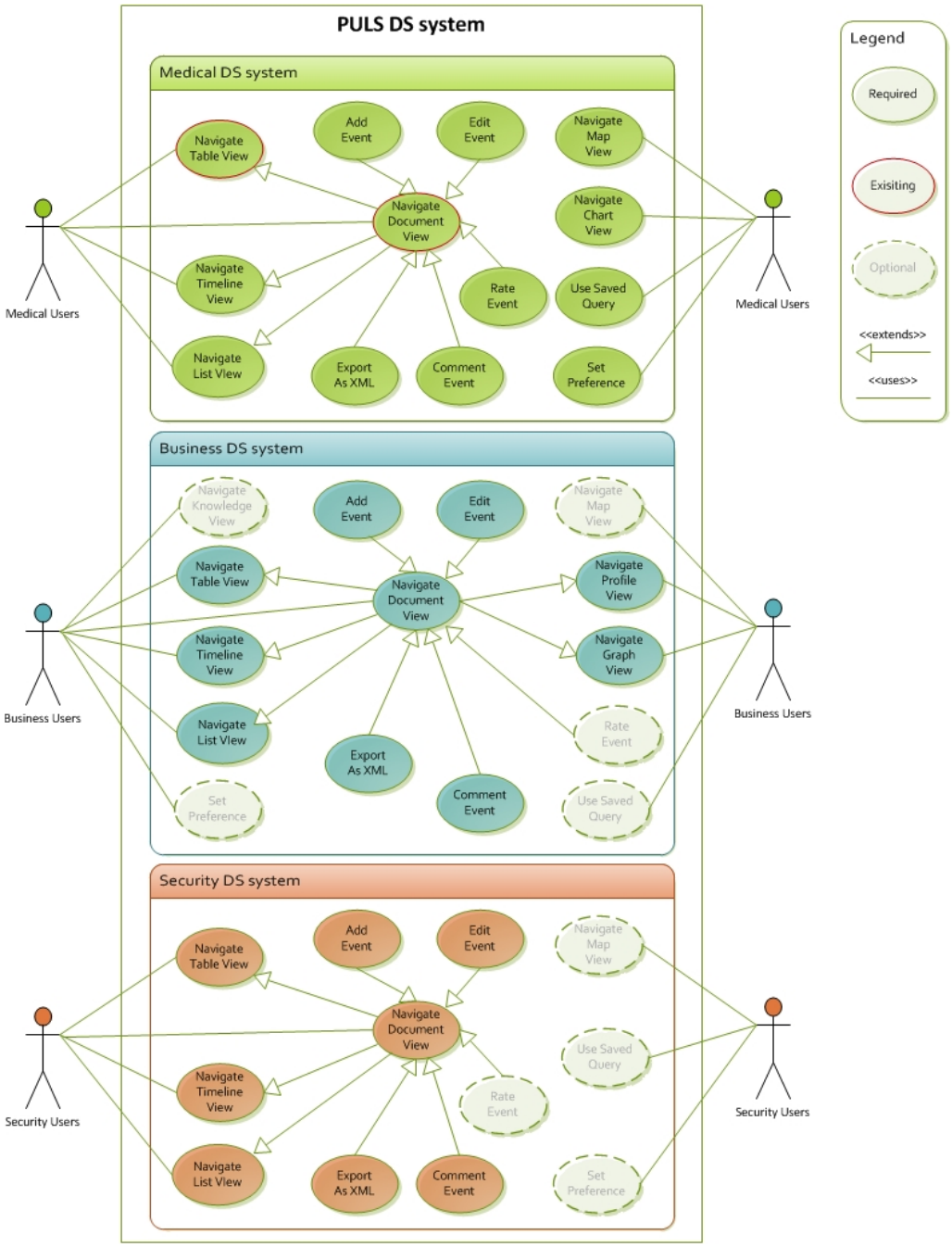


Figure 5: Use Case Diagram for PULS DS system users

### 4.1.4 Class Diagram

The Class Diagrams as shown in Figure 6 and Figure 7 present the static internal structure of PULS IE system and DS system. From them, we may clearly see what objects and how these objects are handled inside PULS system to achieve PULS functionalities described in the previous section.

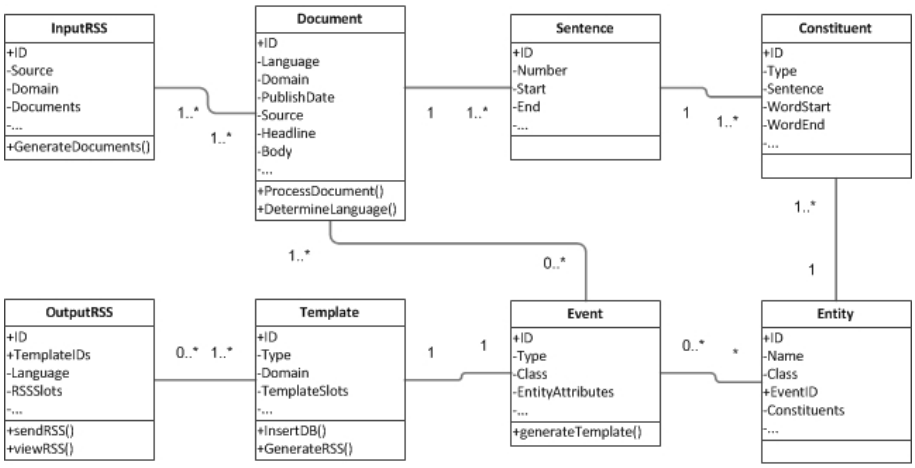


Figure 6: Class Diagram for PULS IE system

### 4.1.5 Activity Diagram

The Activity Diagram as shown in Figure 8 presents us the dynamic view of PULS IE system. It clearly demonstrates the sequential activities that users may do with PULS IE system.

## 4.2 Interface design

Human Computer Interaction (HCI) is a subject of science which is used to research how human being can better communicate with computers. In the past, developers of systems did not pay much attention to HCI as they were expert computer users. However, users of a system who are domain experts may not be computer experts. They need a friendly and easy way to communicate with the systems. Since there is no user interface provided to pure PULS IE users, we concentrate on interface design for PULS DS system. In this section, a draft Interface of PULS DS system

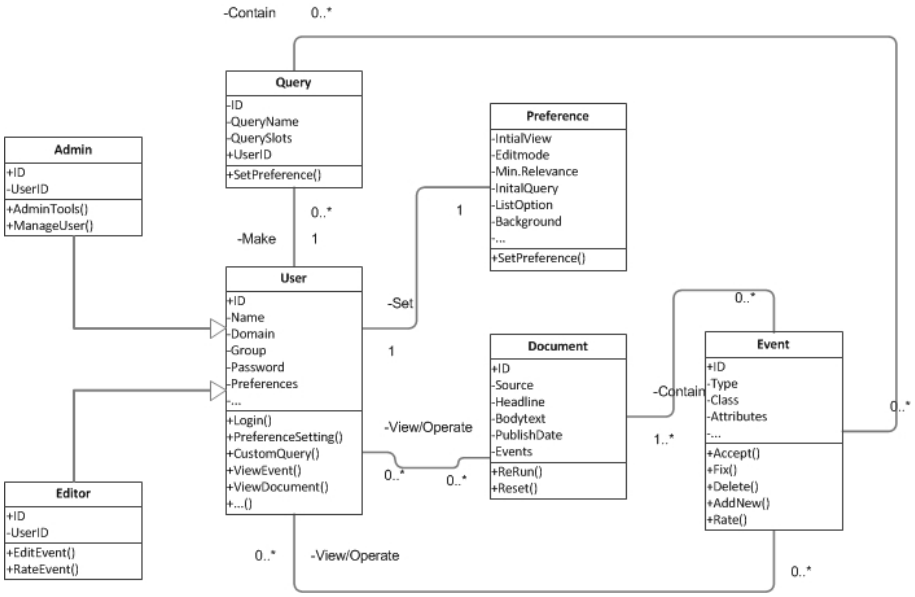


Figure 7: Class Diagram for PULS DS system

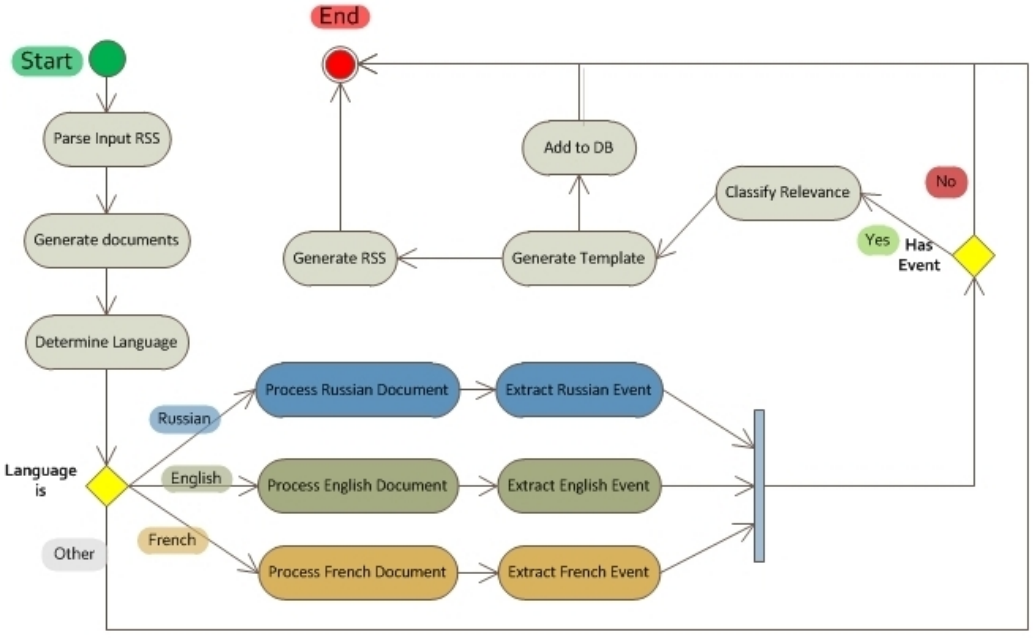


Figure 8: Activity Diagram for PULS IE system

will be presented including layout design, navigation design, content organization design and detailed content display design.

### 4.2.1 Layout design

The layout design can be considered as the most important part of building a Web-based system. If the site looks bad, visitors will not stay long or come back again. Generally, the layout is how features are arranged on the pages. There are many different ways to arrange the contents, and PULS DS system uses the simple standard layout as shown in Figure 9 to display contents throughout the system after several discussions with users.

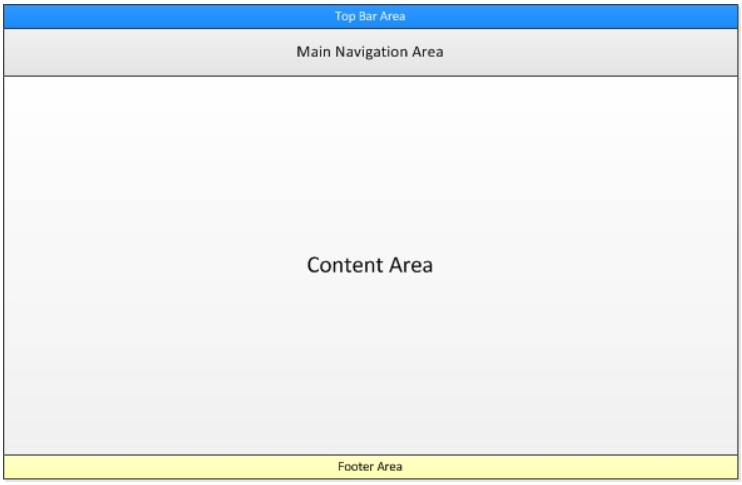


Figure 9: Layout design of PULS DS system

### 4.2.2 Navigation design

*"You can have all kinds of great attractions on your site, but if your visitors don't know how to get to them, they'll just collect dust on the server. Worse yet, if visitors find your site's navigation confusing or convoluted, they'll simply give up and head off to explore the rest of the Web, never to return."* [Tim00] Therefore, a great navigation design is needed for any good website.Navigation design varies somewhat for different types of sites, but they do share the following basic characteristics [RaA95]:

- Minimize travel: any two points should be connected by the shortest and

simplest path.

- Minimize depth: the hierarchy used should have the fewest possible levels.
- Minimize redundancy: multiple paths to the same destination from the same screen should be avoided. This can confuse users about which to choose. In the context of the web however, controlled and well thought out redundancy can improve a site's navigability.

There are many styles of navigation:

- Embedded links: the most basic form of navigation.
- Bread-crumb trail: if you're organizing large amounts of information.
- Navigation bar: Most common, generally usable.
- Tab navigation: When breaking into a few primary categories.
- Site map: One-stop shopping for everything on your site.
- Mix and match navigation schemes for optimal usability.

The choice of navigation styles should be based on the purpose and contents of the website. In general, none of these navigation styles will work for every occasion and the best thing to do is to mix them to achieve a more intuitive navigation.

According to the layout design of the PULS DS system as shown in Figure 9, the navigation design of PULS DS system combines embedded links, navigation bars, tab navigation and site map as described below:

- The top navigation bar which includes links to special advanced pages such as *home page*, *advanced search page*, *databases page* and links to other features users may use, such as *preference setting* or *help* throughout visiting PULS DS system will be provided in the top bar area as shown in Figure 9 and it will be available all the time.
- The main navigation bar will be put in the main navigation area as shown in Figure 9 and it will be always visible. It displays all the primary views such as *table view*, *list view*, *timeline view*, *map view*, etc. as shown in Figure 5 that a member or administrator is allowed to navigate at any time. For different domains, the primary views could be different. For example, profile view is

especially designed for business domain and hence will not be displayed in the navigation bar of medical or security DS systems.

- A bottom navigation bar will be put in the footer area. It contains the links to the project related information such as *about*, *partnership*, *contact*, etc. Also, the link to site map which shows the entire picture of system structure will be integrated into this bottom navigation bar.

- An event-type tab navigation will be used in some primary views for business and security domain to allow users to separately navigate events with different event types. Another tab navigation will be used in user preferences dialog box to separate settings for different purposes.

- Embedded links will be largely provided on almost all page contents. The most common one would be the embedded link to a specific document view which will be placed in the content area of the primary views. While the primary views show us the summary information of many events at once, the specific document view allows users to check the detailed information of an event in a document and possibly update it interactively.

### 4.2.3 Content organization design

Content organization is a general topic in our daily lives. We may find that it is easy to find a specific topic or content in a book since books separate contents into chapters and sections and generally provide us an index for those contents. A good content organization is also important for any website. *"How you arrange the content determines how easy it is to get to every piece of information"* [RaA95]. A good content organization organizes your website in a well-structured manner and benefits the users to better communicate with the website. Based on the book called *Web Page Engineering: Beyond Web Page Design* [Tho98], content on the website can be organized by using four main hypertext organizational schemas: linear forms, hierarchies, grids, and pure Web. Good combination of these organizational schemas is important for making a website easy to use. A mixed hierarchy scheme has been chosen to organize PULS DS system. Similar to the index of a book, hierarchy schema try to organize the content of a website in a structured way which is adopted by most websites. Figure 10 demonstrates the content organization design of PULS DS system. By using the mixed hierarchical scheme, all contents of PULS DS systems are accessible through at most two clicks.

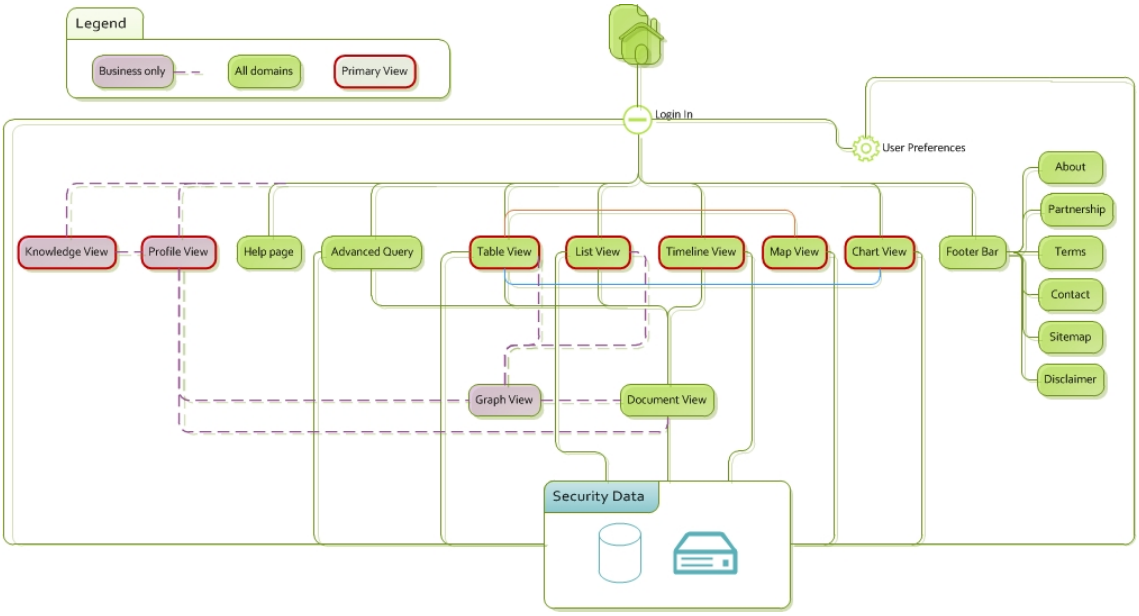


Figure 10: Content organization design of PULS DS system

### 4.2.4 Detailed content display design

As illustrated in Figure 9, the content of different views will be placed in the content area. Different views have different contents as described below:

- **Table view** (Figure 13 in Chapter 5) displays summary information of 20 events in a table on each page. The columns are attributes of events according to the domain. For example, *country*, *disease name*, *total number*, etc. are displayed for medical domain while *company name*, *business activity type*, *industry sector*, *description of business activity*, etc. are selected to display for business domain. Each column header is the attribute name and is click-able for sorting, and each column is also search-able by using the search box under the attribute name.

- **List view** (Figure 15 in Chapter 5) contains 10 groups of similar related events (e.g., same disease outbreaks in same country within a short period) displayed as a list on each page. Each group is formed by the group name header, information of documents belonging to this group and all events extracted from those documents. A search bar is provided at the top to search required groups by grouping attributes.

- **Timeline view** (Figure 16 in Chapter 5) shows events' label (e.g. disease name for medical events or company name for business events) chronologically. Each bullet represents an event and the different horizontal positions of the bullets indicate the time differences of the events.

- **Map view** (Figure 17 in Chapter 5) visualizes the occurrence and frequency of events geographically. A marker in a specific location indicates that there are some events happened in that location and the size of the marker will be proportioned according to the number of events.

- **Advance query** looks similar to table view. There will be a advanced search box under the table which allows users to search by more attributes.

- **Document view** (Figure 14 in Chapter 5) contains a news article text box on the left, an information/edit box for event on the right, a comment box under the article text box followed by the related events box. The information/edit box can be toggled by clicking the edit-mode button.

- In business DS system, **profile view** (Figure 19 in Chapter 5) uses a number of interactive IV tools to visualize the profile of a certain business entity (company or business sector) and allow users to interact with these tools. A search box will be provided at the top for users to search for an entity. Tag cloud will be chosen to display the relationship among this entity and other entities; list will be used to present all events related to this entity; pie chart will be utilized to demonstrate proportionally the information of a specific attribute (i.e. country, event-type) of all events related to this entity; and line chart will be adopted to visualize the frequency of related events occurred chronologically.

- In business DS system, **graph view** (Figure 18 in Chapter 5) uses interactive graph to visualize the relationship among business entities (companies, persons, products). The graph node represents an entity and the edge between each two nodes describes an event. Users may navigate the graph easily by zooming in/out; highlighting all entities directly related to a specific entity; searching for a certain entity; clicking an entity node to expand the graph using clicked entity as the center node; or clicking an edge to go to the document view that contains the event represented by the edge.

- In business DS system, **knowledge view** uses several lists to present the knowledge of relations among business entities. A search box will be provided

at the top for users to search any business entity (e.g. company, person, sector, position, etc.). The search result will contain all accumulated knowledge of relations among this entity and other entities which will be presented by different lists grouped by relation types.

- **Preference setting** (Figure 20 in Chapter 5) will be a pop up box which can be popped up on top of any view described above. Preference settings are classified into categories and each category has a separate tab box which contains the settings of that category. For example, the document view tab contains the settings for document view (e.g., toggle of edit mode initially when document view is loaded, and the background image of the document view).

## 4.3 Database Design

Elmasri and Navathe have mentioned in a book [ElN03]: *"a data model represents collection of concepts that can be used to describe the structure of a database. The structure of database includes data types, relationships and constraints that should hold on the data."*

In order to design database of PULS system, a data model will be constructed first using Entity-Relationship modelling which is one of the most widely-used approaches for data modelling; the ER Diagram will then be mapped to a relational database.

### 4.3.1 ER Diagram

According to the class diagram designs as shown by Figure 6 and Figure 7 in Section 4.1.4, I have drawn the ER Diagram for the PULS as illustrated by Figure 11. This diagram shows the integrated entity relations for both PULS IE system and PULS DS system. Since currently we only store the extraction results and corresponding information, not all classes described in the class diagram designs are needed in the database.

### 4.3.2 ER Diagram to relational database mapping

Elmasri and Navathe have analyzed in their book seven rules used for mapping ER diagrams to relational databases. As the ER Diagram for PULS does not need to

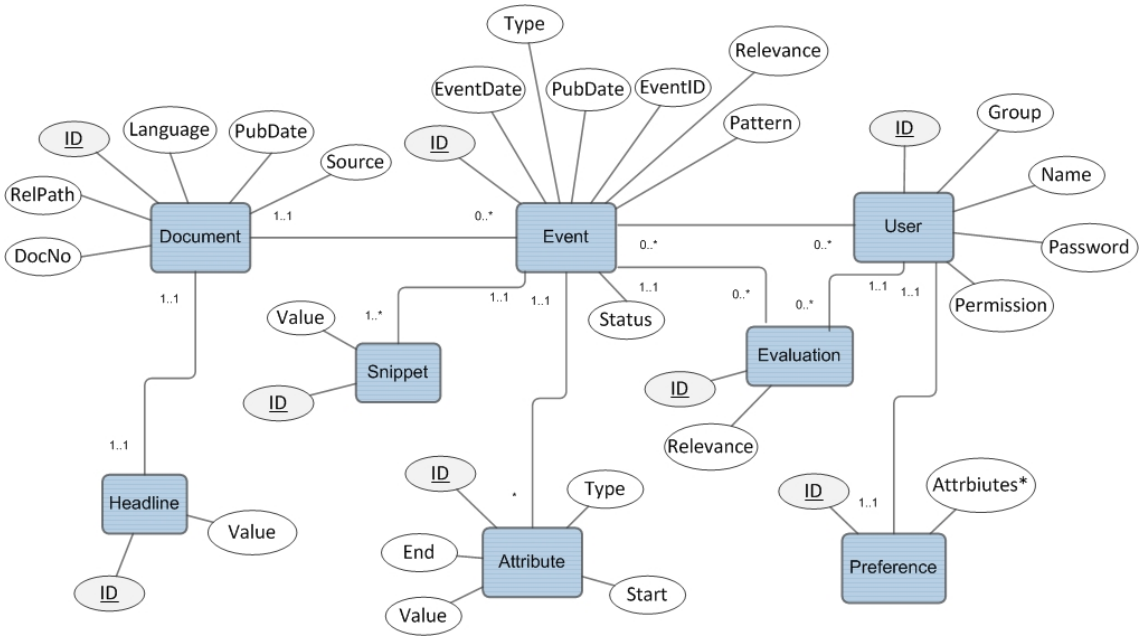


Figure 11: Entity Relational Diagram for PULS

use all 7 rules, 4 rules are used to map the ER Diagram to a relational database as shown below [ElN03]:

- Rule 1: For each entity E in the ER schema, create a relation R; include all the attributes of E in the relation R; choose one of the key attributes of E as the primary key for R. For example: for entity Document in ER Diagram, We have created a relation called *Document*; Include all attributes of Document (ID, DocumentNo, Language, FilePath, Source) in the relation Document; choose ID as the primary key for Document.

- Rule 2: For each one-to-one relationship, use foreign key or merged relationship or cross-reference approach to establish the relations. For PULS, we choose foreign key approach. For example, for one-to-one relationship between relation Document and relation Headline, we have included the primary key of Document as the foreign key called *documentID* in the relation Headline.

- Rule 3: For each one-to-many relationship in the ER schema, include the primary key of the relation S (that represents the participating entity at the one-side of the relationship) as the foreign key in the relation T (that represents

the participating entity at the many-side of the relationship). For example: for one-to-many relationship between entity Document and Event in ER Diagram, We have included the primary key of relation Document as the foreign key called *documentID* in the relation Plus.

- Rule 4: For each many-to-many relationship in the ER schema, create a new relation R; include the primary keys of the both relation S and relation T (that represent the participating entity at the two many-side of the relationship) as the two key attributes in the relation R. Then apply rule 2 since now both relation S and Relation T have the one-to-many relationship with the new relation R. For example, there is a many-to-many relationship between entity User and Event in PULS ER Diagram since the users can rate/edit more than one event and one event can be rated/updated by more than one users. We have included the primary key of both relation User and Event called *userID* and *eventID* as the foreign keys of the new relation UserEvaluation.

By applying these four rules above, We have mapped the ER Diagram to PULS relational database.

## 4.4 Chapter summary

In this chapter, requirements for PULS have been translated into actual designs. Functionalities of the PULS IE and DS system designed using UML have been presented first, followed by the user interface designs of PULS DS system. Finally, database design of PULS has been presented. In the next chapter, detailed implementation to realize these designs will be presented.

# 5 Implementation

In previous chapter, requirements for PULS have been translated into actual designs. This chapter will focus on the realization of those designs by implementations. The implementation technology and environment will be introduced first followed by the detailed implementation of those designed functionalities and user interface. Finally, the implementation of PULS database will be presented.

## 5.1 Implementation technologies and environment

### 5.1.1 Programming environment

The original PULS system was developed using Lisp in Linux. There are a number of advantages of using Lisp for NLP as stated in Section 2.3.1 which makes it easy for us to create any kind of desired objects, achieve those proposed functionalities as designed in the previous chapter, and test them.

In order to facilitate the programming and debugging, we choose to use Steel Bank Common Lisp (SBCL), GNU Emacs and The Superior Lisp Interaction Mode for Emacs (SLIME) for interactively implementing the required functionalities.

SBCL is a high performance Common Lisp compiler. Besides the compiler and runtime system for ANSI Common Lisp, it provides an interactive environment including a debugger, a statistical profiler, a code coverage tool, and many other extensions [SBC11].

GNU Emacs is an extensible, customizable text editor for Lisp programming with extensions to support text editing [GNU11] while SLIME is a Emacs mode for Common Lisp development. With slime-mode, we can do the following things conveniently [SLI11],

- code evaluation, compilation, and macroexpansion
- online documentation
- definition finding
- symbol and package name completion
- automatic macro indentation

- cross-reference interface

For database communications, we use CLSQL, a SQL database interface for Common Lisp which is compatible with AllegroCL, Lispworks, SBCL, CMUCL, and OpenMCL [CLS11].

For Web-based PULS DS system, we use Hunchentoot which is a web server written in Common Lisp and also a toolkit for building dynamic websites. As a stand-alone web server, Hunchentoot provides all we need to host the PULS DS system like automatic session handling, logging, customizable error handling, and easy access to GET and POST parameters [HUN11].

All of these tools stated above form a pure Lisp implementation environment which greatly facilitates the communications (e.g. sharing code, passing objects) among different PULS packages.

### 5.1.2 XML used in PULS

Since PULS is one key component of a large distributed news surveillance system, XML was largely used to communicate among distributed components of the system. PULS receives RSS files which are essentially XML formatted plain text from our partners' systems. These RSS files contain the semi-structured news articles (including source, title, published date, content, etc.) collected by our partners' systems from the Web. The outputs of processing and extracting information and knowledge from these data by PULS are also converted to the RSS files which are then sent to other partner systems for data mining purpose.

Besides data exchanging, these RSS files which store the original and processed information are also used by PULS for backup purposes and are kept as one kind of data resources in the data module of DSS.

### 5.1.3 Other technologies

Crontab, Fetchmail, Shell Script, Apache Tomcat, Servlet, etc., are used to fetch RSS feed, invoke PULS IE processes, send out RSS results in the back-end.

JavaScript/Jquery, Json, AJAX, CSS are used in the PULS DS system to facilitate the interactive communications between users and DS system.

Java, Weka toolkit [Wek12, HFH09] and machine learning technologies are used for

building learning classifiers to improve PULS system in several ways.

## 5.2 Functionalities implementation

Since the detailed designs using UML which clearly interpret the Functional requirements have been made, the implementations of those functionalities goes smoothly. All required IE functionalities as shown by Figure 4 and Web DS features as shown by Figure 5 have been successfully implemented using technologies described in the previous section.

### 5.2.1 PULS IE system

**PULS security IE system** PULS IE system now contains the new sub-system for cross-border security domain. It receives and stores the RSS feed file from security partner system continuously; processes approximately 500 English and 100 Russian documents in the RSS file daily; extracts around 200 cross-border security events from these documents; classifies them into migration, human trafficking, smuggle or crisis events and possibly their sub-types; generates the response outputs and adds them into the database (FR1, FR3). More details are described in [ADP13, DVK11].

**PULS medical IE system** For medical domain, we have integrated the French pipeline in medical IE sub-system to process French documents and extract the disease outbreak reported in French (FR2). Around 15,000 documents are processed daily by either English pipeline or French pipeline according to their languages and approximately 1% or them contain real medical outbreak facts.

**PULS business IE system** For business domain, we have built the interface to accommodate the new business source. Now, we process over 5,000 news articles received from two business sources everyday, extract facts from them and add the extracted business events to two separate business databases. For each of them, we are able to classify the events into investments, acquisitions, new products, posts, marketing, layoffs, contracts, orders, ownership, mergers according to the business activity specifications (FR4, FR6).

While implementing these new functionalities, we have been also working on improving the quality of IE extraction results determined by the F-measure for all domains.

We have been accumulating knowledge into our knowledge base for achieving experimental optional requirements (FR7, FR10), such as identifying business sector of a business activity; merging similar events from same or different documents; distinguishing persons with political post and business people, etc. We have also expanded our validation suites of all three domains for better evaluation.

**A walkthrough example** A walkthrough example of PULS medical IE system is demonstrated by Figure 12. It is applicable in other domains as well.

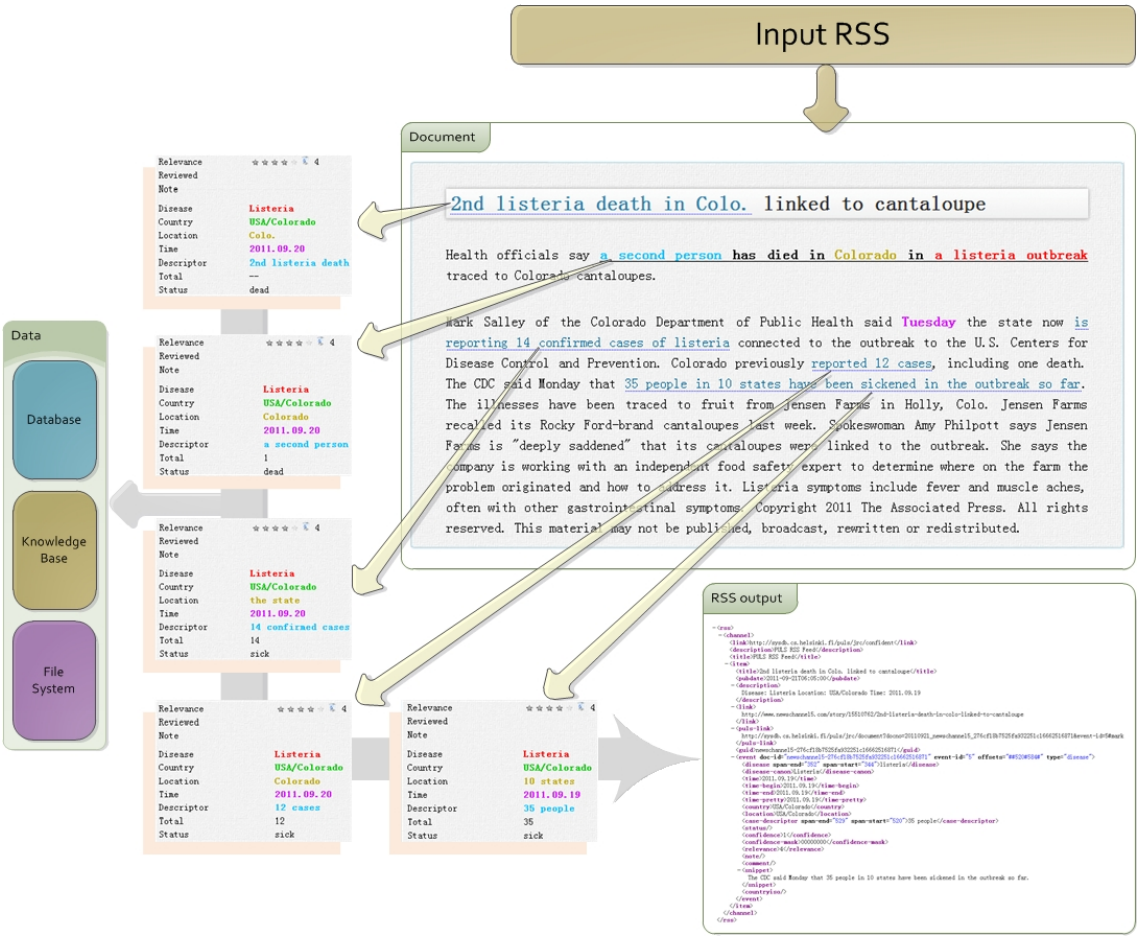


Figure 12: A walkthrough example of PULS medical IE system

From this example, we can clearly see the whole process of PULS IE system for each domain which can be divided into the following steps,

- PULS receives RSS input files containing the semi-structured news articles

(including source, title, published date, content, etc.) and store them in the file system.

- RSS are decomposed into separate document files stored in the file system, and their information recorded in the database and knowledge base.
- Documents are processed and required facts are extracted from the documents by PULS IE system. One document may not have any fact or it may contain several facts with pre-specified attributes, as shown in Figure 12.
- Extracted facts are inserted into the database. Additional information from extraction is added to the knowledge base.
- The output RSS files are generated from the extraction results which are then sent to the partner system users.

These steps are implemented exactly according to the requirements specified in Table 25 and functionality designs presented in Figure 4 and Figure 8.

### 5.2.2 PULS relevance classifiers

We have implemented our relevance classifiers (FR9) for all three domains using the WEKA toolkit [Wek12, HFH09], which provides a collection of machine learning algorithms. For each domain, we have accumulated a number of manually rated training/testing data through our newly implemented edit function (FR18). The relevance of any extracted event, which was rated by users based on the pre-defined criteria (Table 2), falls into 6 classes from zero to five. User evaluated data is split into three classes in our classifiers; high-relevance (relevance scores 4 and 5), low relevance (relevance scores from 1 to 3) and irrelevant (relevance score of 0). We train our classifiers using these three classes instead.

In order to build the three-way classifier, we train two types of binary classifiers: the high-vs-low classifier separates between events labeled 4–5 and 0–3. The zero-vs-rest classifier separates the zero-relevance (i.e., completely useless) events from the rest. These two classifiers are combined in a sequential order. We first use the zero-vs-rest classifier to collect zero-relevance events and then use the high-vs-low classifier to separate high-relevance events from low-relevance events. These events are therefore falling into three classes, i.e., high-relevance (relevance scores 4 and 5), low relevance (relevance scores from 1 to 3) and irrelevant as expected.

The features we use in relevance classifiers fail into two groups, discourse features and *lexical features*.

**Discourse features** operate on information about the article such as the number of events found in the article, the positioning of the events in the document, the compactness of the placement of the event's attributes [BaB97, HYG02], and the recency of the events' occurrence. Table 3 shows some examples of discourse features.

| **Layout features** |
|---|
| Event trigger in header or headline |
| Any event trigger found in header/headline |
| Trigger's relative location in document |
| Actor in trigger/header/headline |
| Country in trigger/header/headline |
| Document length |
| **Compactness** |
| Distance from trigger to actor |
| Actor found before end of trigger sentence |
| Number of unique countries/actors in trigger sentence |
| Number of unique countries/actors until end of trigger sentence |
| Number of unique countries/actors found in document events |
| Event contains a valid country/actor |
| Is content repeated in the header/in the document |
| Number of events found in document |
| **Time** |
| Event has time of occurrence |
| Distance between trigger sentence and mention of event date |
| Time difference betwen publication date and event start/end date |
| **Low relevance indicators** |
| Is blacklisted data found in Headline/Header/Document |
| Number of *harm* events in the document (medical domain) |
| Victim count |
| Is victim named |
| Is illness unspecified |

Table 3: Examples of discourse features

**Lexical features** for an event consists of bags of words in the trigger sentence, and

in the sentences immediately preceding and following the trigger sentence.

Implementation details of relevance classifiers are described in [HVE11, HVD13]. According to the users' feedback, such attribute of event has greatly assisted them to filter out irrelevant events that users do not want to spent time on (FR9).

### 5.2.3 PULS DS system

All required dynamic views of PULS DS system described by FR11 - FR22 have been successfully implemented using Lisp and Javascript. While Lisp programs generate static Web pages on the server side in responding to users' queries and navigations, Javascript functions provide dynamical user interfaces on the client side which interactively communicate with users.

**Table view** For all domains, *table view* (FR15) has been improved to be more interactive which now supports sorting by any column, Regular Expression and advanced searching, surveillance (events with high relevance) and complete view switching, etc., as shown by Figure 13. The implementation detail is given in Table 30 in Appendix 3.


Figure 13: Table view

**Document view** *Document view* (FR18) has also been extended to allow users to make comments to, add, edit, rate or export as xml any event in the document and provide possible related events links as shown by Figure 14. The implementation detail is given in Table 31 in Appendix 3.

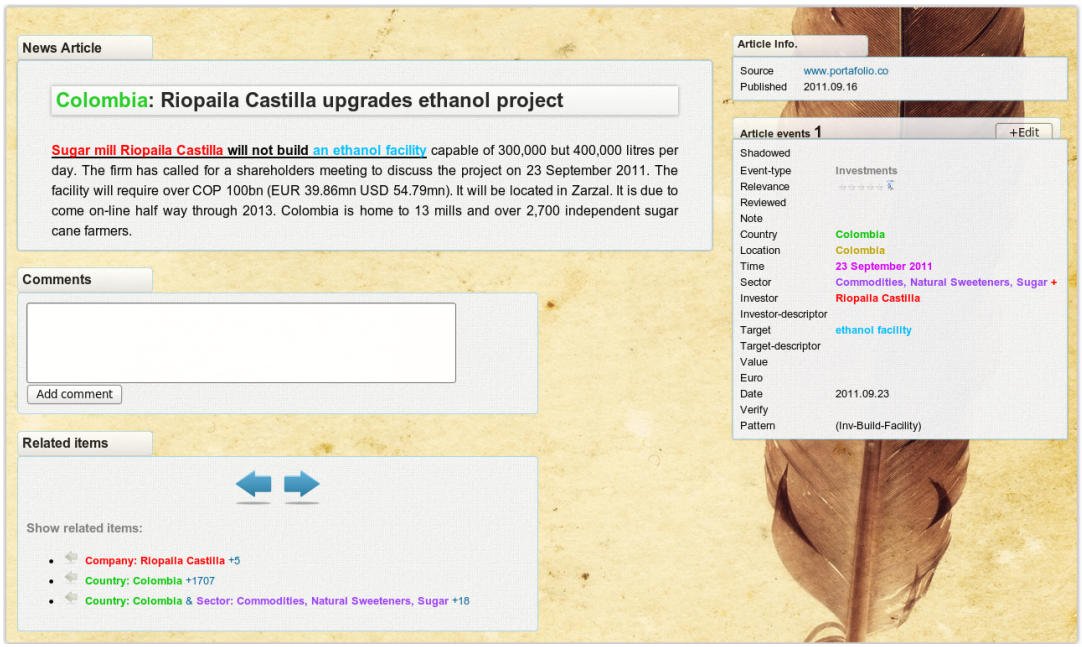


Figure 14: Document view

Besides these improved views, *list view* (FR11), *timeline view* (FR12) and *map view* (FR13) have been integrated into the DS system for all domains. The implementation details are presented in Table 32, 33 and 34 in Appendix 3.

**List view** *List view* displays related events in groups (Figure 15).

**Timeline view** *Timeline view* displays events chronologically. Figure 16 shows an example of events in *Electronics, Telecommunications Terminals, Telephones* sector displayed in the *timeline view*.

**Map view** *Map view* visualizes the occurrence and frequency of events geographically (Figure 17).

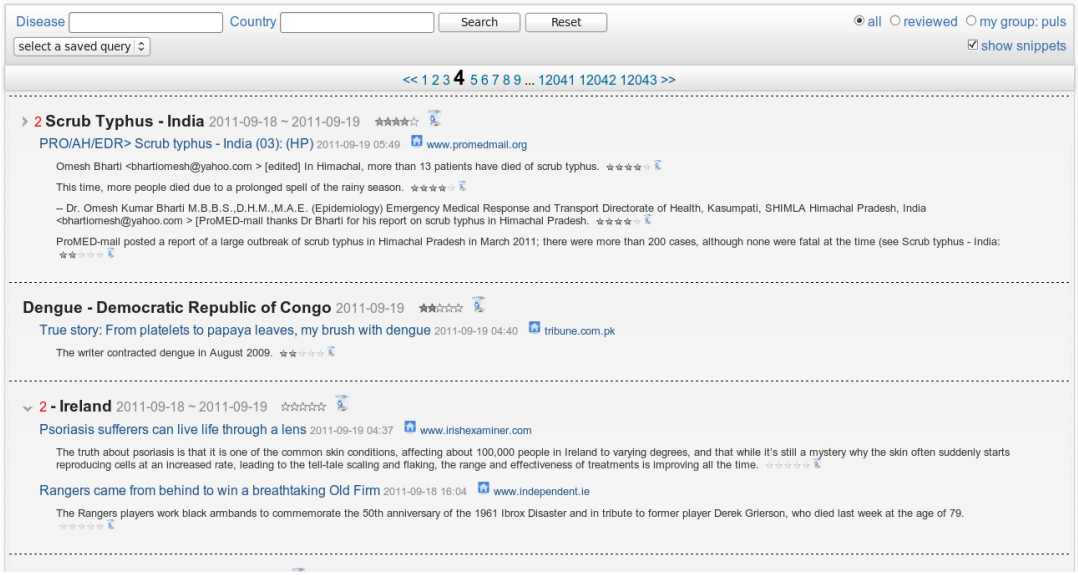


Figure 15: List view

**Other features** Graph view (FR13) (Figure 18), profile view (FR16) (Figure 19) and knowledge view (FR22) have been provided exclusively in business DS system. The implementation details to achieve these features are shown in Table 35 and Table 36 in Appendix 3.

In addition, users can customize their preferred view using the new preference setting feature, a pop up box as shown by Figure 20. Preference settings are classified into categories and each category has a separate tab box which contains the settings of that category. Uses may also save their customized queries here which can be used in table or list view later.

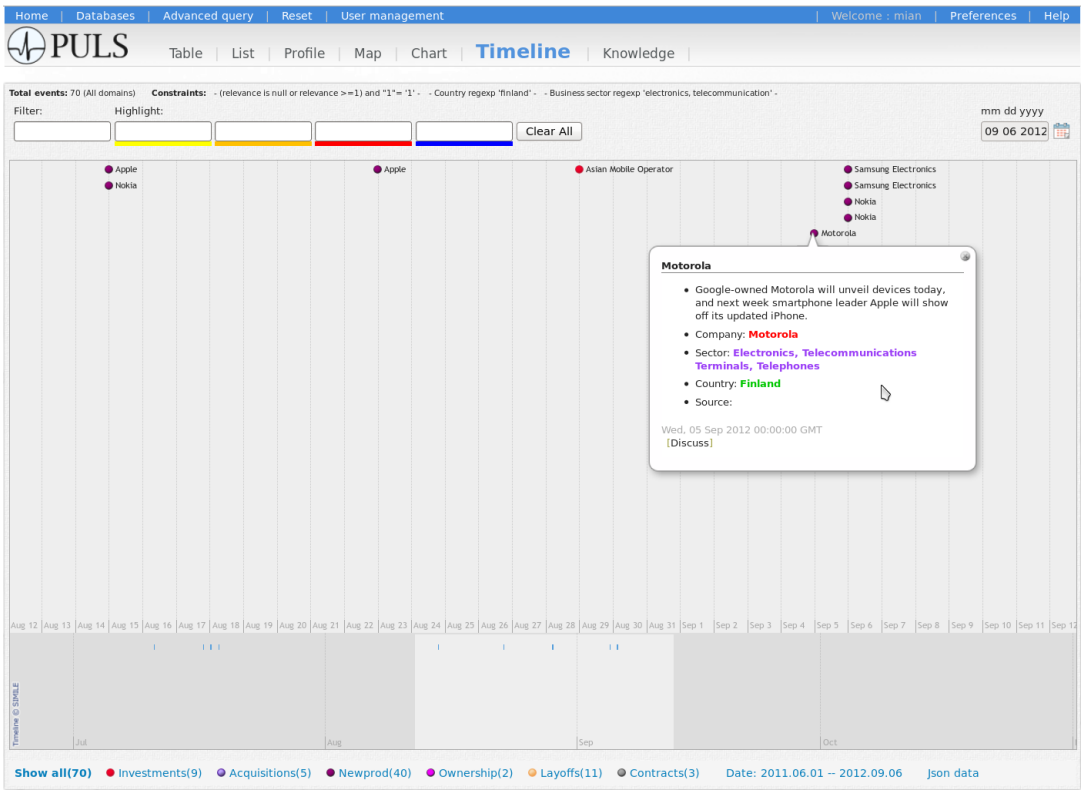


Figure 16: Timeline view

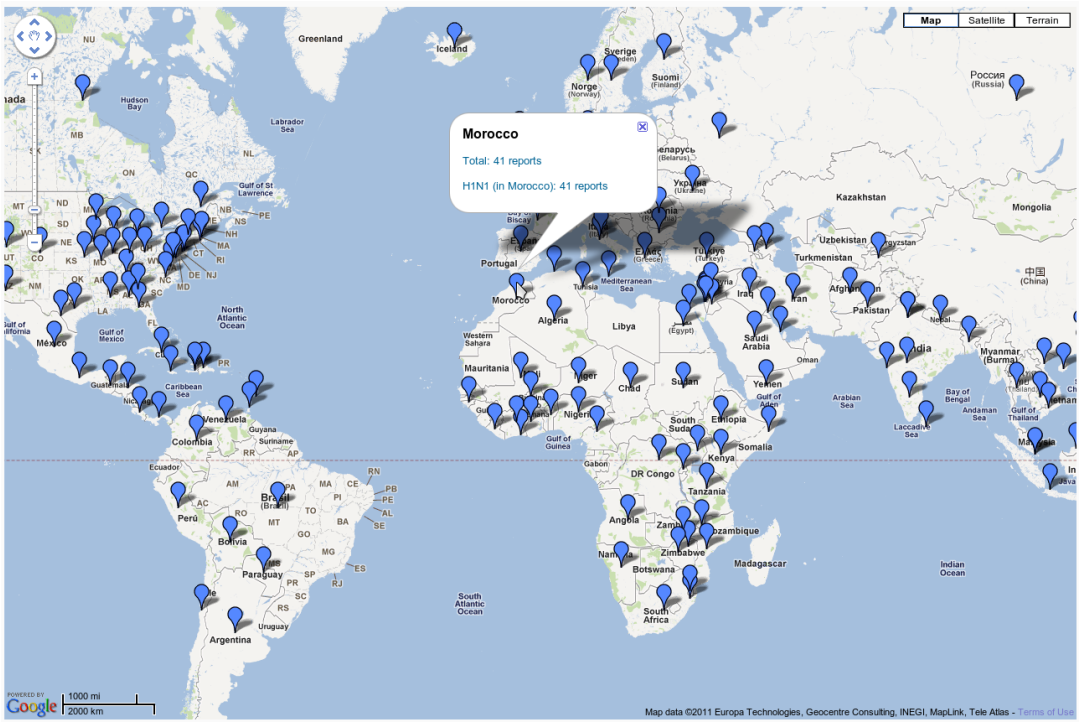


Figure 17: Map view

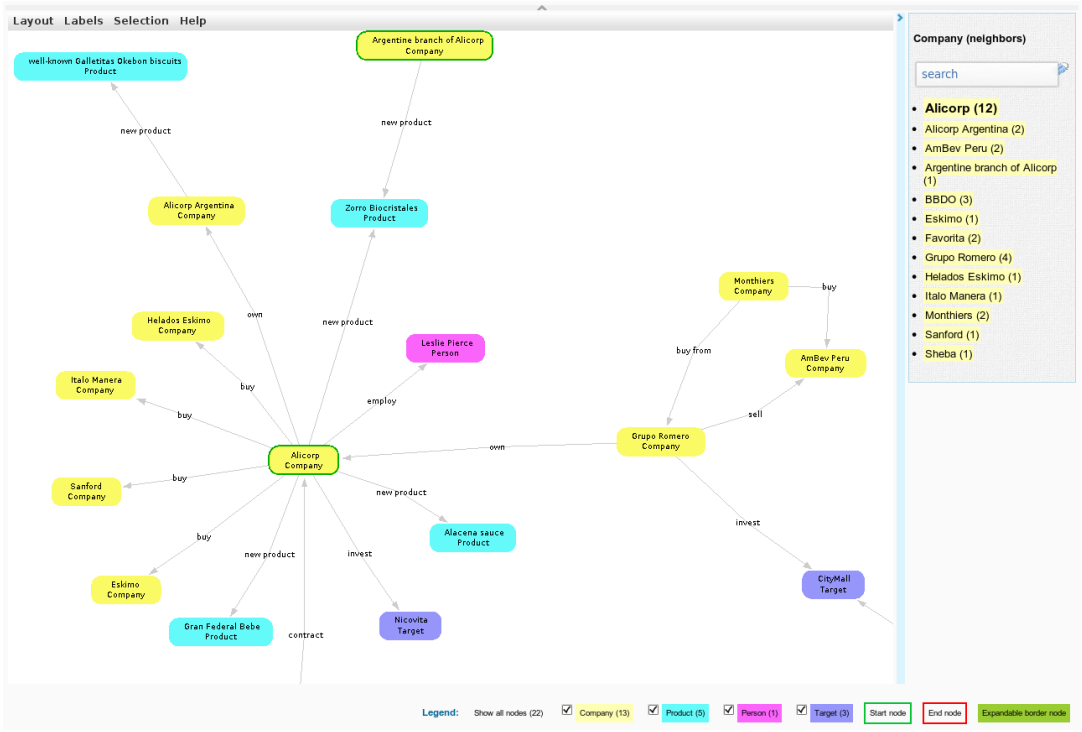


Figure 18: Graph view

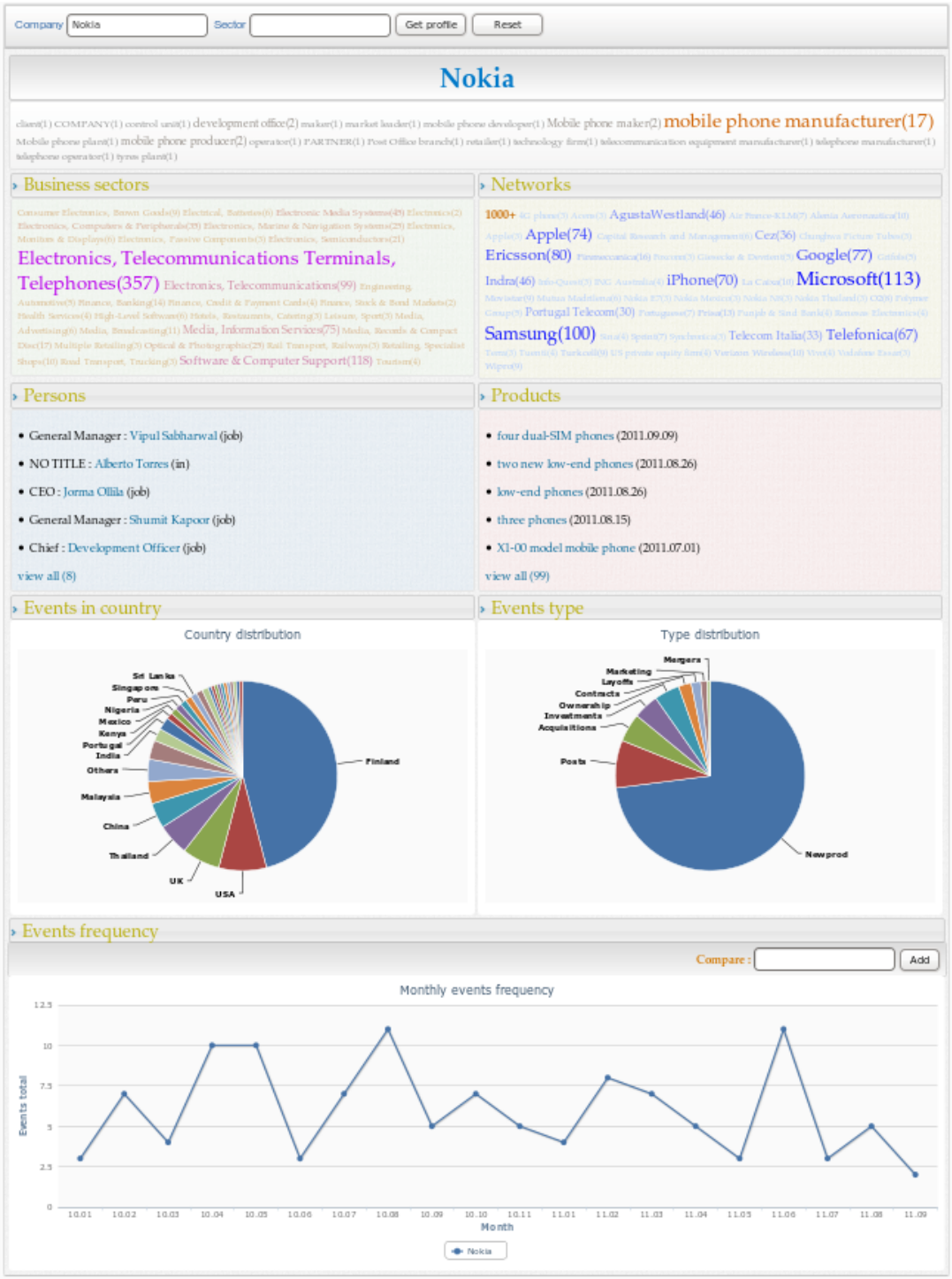


Figure 19: Profile view


Figure 20: Preference settings

## 5.3 Database implementation

Based on the clearly defined ER Diagram as shown in Figure 11 and its corresponding PULS relational database as shown by Figure 11, we have implemented PULS database easily.

MySQL database has been chosen for our database because of its high performance, high reliability and ease of use [MyS12]. As described before, we extract events from news belong to medical, business and cross-border security domain. For each domain, we have created one MySQL database and the same set of tables in the database as illustrated by Figure 11.

- Table *document* stores information of news articles.
- Table *plus* contains all information related to extracted events.
- Table *evaluation* is used to store all relevance evaluations made to the event.
- Table *user* holds the status and information of PULS users.
- Table *event_group* groups facts from plus by specified attributes like disease and country for medical domain to serve the *List View*.

These database tables represent entities in PULS system and they have been converted into Lisp objects in the program. We have then implemented models for all domains on top of CLSQL to handle these objects and easily interact with database. The relations among these entities as illustrated in Figure 11 have been also implemented in MySQL database and LISP database models. Additionally, common query interfaces have been implemented based on CLSQL as well so that PULS system is capable of communicating with database conveniently and directly, without using SQL commands.

## 5.4 Chapter summary

In this chapter, the implementation technologies and environment are introduced first. The implementation details of functionalities and user interfaces for both PULS IE system and PULS DS system are presented one by one. Finally, we see how the PULS database has been realized to serve those functionalities. In the next chapter, evaluation and testing of PULS system will be presented.

# 6 Evaluation and Testing

Evaluation and testing should be performed in all stages of development. For example, at the requirement stage, we need to validate the requirements with actual users before creating the requirement specification; at the design stage, we need to evaluate whether the design has correctly and completely accommodated the requirements identified in the requirement stage, etc. Further, a number of approaches should be used to test and evaluate whether the newly implemented functionalities of PULS can work correctly and effectively.

Therefore, the purpose of the evaluation and testing stage mainly includes:

- ensuring all the functionalities of PULS can work correctly;
- ensuring PULS can meet the users' requirements specified.
- evaluating the performances of all functionalities using appropriate evaluation metrics.

In this chapter, internal testing will first be presented which aims to ensure all the functionalities of PULS can work properly. External testing with the attempt to ensure that PULS can satisfy the actual users' needs will then be presented. Finally, evaluations of the main functionalities will be presented.

## 6.1 Internal testing

Internal testing is trying to ensure all functionalities implemented can work adequately and correctly. We follow several steps to perform the internal testing. Firstly, unit testing which tests whether each unit of PULS is working will be presented. Secondly, we will concentrate on the integrity testing which tests whether PULS can work properly after all units have been integrated. Then, browser and resolution testing with the attempt to ensure that PULS can function well regardless of the configuration of users' system will be presented. Finally, delivery testing will be presented.

### 6.1.1 Unit testing

Unit testing can be considered as the fundamental testing of any system. It tries to test the smallest units of the system and tries to find any error of the unit itself and

error users may encounter while using them. Since the programmers know a unit best while coding that unit, performing the unit testing during coding is very useful and effective.

As illustrated in Figure 4, *PULS IE system* is formed by three sub-systems. Each sub-system serves a domain and it contains a number of use cases. Every use case in Figure 4 can be considered as a component unit in PULS IE system and it needs to be tested first by itself. The testing includes:

- **Input testing**: the required input is acceptable.
- **Output testing**: the correct output can be produced.
- **Error handling testing**: all implemented error handling capability can function well.
- **Other functionalities testing**: all other functionalities work as designed.

Table 37 in Appendix 4 shows one example of the unit test for **Generate Template**. All units in PULS IE systems have been tested according to this table.

In order to facilitate the unit testing, we have made a testing script for each unit. After the unit has been implemented or changed, the testing script is then used to test the unit as described above.

For PULS DS system, every use case illustrated in Figure 5 is treated as one unit. The unit testing for them includes:

- **Visual testing**: such as content display, navigation display, font resizing, window resizing, etc.
- **Access control testing**: if the unit can only be used by members or administrators, check the access control capability can function well.
- **Error handling testing**: all implemented error handling capability can function well.
- **Interaction testing (data testing)**: if the unit will interact with database, check whether the communication is correctly performed.
- **Other functionalities testing**: all other functionalities work as designed.

Table 38 in Appendix 4 describes an example of such unit testing performed for all units in PULS DS system.

Rather than using testing script, the unit testings for PULS DS system are performed manually every time a new function has been implemented in the unit since most of the functions require interaction between PULS DS system and users. Therefore, a testing guideline for each unit which contains all the types of testing as shown above has been made.

### 6.1.2 Integrity testing

Integrity testing has been performed in order to ensure PULS can run well after all units have been integrated together.

For **PULS IE system**, the integrity testing ensures the whole IE process pipeline for each domain as demonstrated by Figure 8 works as expected when putting all PULS IE units together.

For each domain, we have built the testing script which contains the sequential tests for all units and runs them one by one. This script is run every time when any unit has been modified. The integrity testing includes the following steps,

- **Parse input RSS**: Testing script parses a testing RSS input file containing a number of news articles (including source, title, published date, content, etc.).
- **Generate documents**: The RSS file is decomposed into separate document files stored in the file system and the meta-data of these documents is recorded in the database and knowledge base.
- **Determine language**: The language of the document is determined by a language classifier and information is stored in the database.
- **Process document**: Depending on the language, the document is processed by different **process document** units and events are extracted from the document.
- **Generate template**: Each event extracted generates one record in template format in the response file.
- **Add to DB**: One event in template format is added as one record into the database.

- **Generate RSS**: The result is converted to specified output RSS format for PULS IE partner users.

The final output of the PULS IE integrity testing is therefore the events recorded in the database and the output RSS file. By comparing events and their detailed attributes with the expected results, we are certain that the whole pipeline is working.

For **PULS DS system**, the integrity testing aims to make sure all links are correct and the interlinked pages can work fine.

All links of PULS DS system have been tested and they are correct according to the content organization design as shown by Figure 10. A special link testing has been performed to ensure that the help documentation can function well (users will be automatically redirected to the help page which contains the help for functionalities provided on the user's current visiting page if they click the link "help").

The inter-operating pages of PULS DS system mean that one page can be only initiated from another page. For example: search result page can only be initiated if users have used the search box to search for certain content. The information related to that search will be passed from the initial page to the search result page. All inter-operating pages include:

- table page $\rightarrow$ table search result page
- list page $\rightarrow$ list search result page
- table search result page or list search result page $\rightarrow$ profile page or map page or timeline page
- document page $\rightarrow$ corresponding related items
- document edit page $\rightarrow$ updated document page
- profile $\rightarrow$ database communication page
- corresponding related items on document page $\rightarrow$ communication page
- relevance update system $\rightarrow$ database communication page
- knowledge page $\rightarrow$ profile page
- preference setting box $\rightarrow$ corresponding updated pages

Various testing data have been used to test whether the cooperations between the pages are correct and finally make sure all cooperative pages perform as expected.

## 6.2 External testing

External testing is trying to ensure that PULS can satisfy users' actual requirements in both functional point of view and non-functional point of view. In this section, a feature checklist which checks whether PULS has satisfied users actual requirements in the functional point of view will first be presented. Usability testing, which aims to confirm that users of PULS can be totally satisfied with non-functional requirements like efficiency, effectiveness, etc., will then be presented.

### 6.2.1 Feature checklist

As mentioned before, all requirements identified at the requirement stage were elicited from the client's view point; those requirements have then been transferred into actual designs in the design chapter; finally, those designs have been realized in the implementation chapter. Feature checklist is therefore to check whether new functionalities of PULS have been implemented correctly and properly according to those designs and requirements. One example of feature checklist is shown in Table 39 in Appendix 4 and feature checklist for all new functionalities of both PULS IE system and PULS DS system specified in Table 25 and Table 26 in Appendix 2 have been performed according to this table.

We can see from Table 39, a number of new features have been successfully implemented in order to meet the functional requirement FR15. These features try to facilitate users to get more information easier and quicker. Further, they provide more interactive ways for users to communicate with PULS DS system in table view. Technically, they do satisfy the requirement specified in FR15. But whether these features can meet users' actual needs should be answered by the following section.

### 6.2.2 Usability testing

Usability testing ensures that users of PULS can be totally satisfied with non-functional requirements like efficiency, effectiveness, etc. In order to achieve usability testing, we need to first decide which technology is most suitable for PULS to perform the usability testing. Two most widely used technologies are known

as cooperative evaluation and questionnaire. While cooperative evaluation is more useful if only a small number of users are available for testing, the questionnaire is more suitable for a large number of users. Since PULS system has a small number of users, cooperative evaluation has been chosen to perform the usability testing.

According to *Cooperative Evaluation: a run-time guide* [MWH93], *"Cooperative Evaluation is a procedure for obtaining data about problems experienced when working with a software product, so that changes can be made to improve the product."* We can divide the cooperative evaluations into three steps as shown below which reflect the main activities of preparing and running an evaluation session.

**Recruit users**: in this step, we need to define the target user population and then invite users to perform the usability evaluation. As mentioned before, users of PULS system can be classified into two groups, i.e. the users of extraction results from the PULS IE system and the users of PULS Web-based user interface. The first group of users come from the partner systems which receive the extraction results as XML files from PULS. The second group of users navigates the PULS user interface to view events they are interested in and use it as a surveillance tool for disease outbreak, business activities, etc. Since PULS DS system is a Web-based application, all Internet users can be considered as potential DSS users. According to the suggestions about how to recruit users [MWH93], several users from PULS partner systems who can represent all DSS users have been invited to perform the cooperative evaluation for PULS.

**Prepare tasks**: selecting the right tasks is vital for the success of cooperative evaluation. A number of meetings have been arranged to perform the cooperative evaluation. Some meetings were held when most functionalities of PULS have been implemented and a final meeting was held after all functionalities have been implemented. In this step, a list of tasks has been prepared for each meeting. The purpose of those tasks are aiming to test all features of PULS implemented and to obtain feedback from users while they are performing those tasks.

**Interact and record**: during this step, users test PULS according to the list of tasks defined in the previous step. When users were testing the PULS, I have recorded all errors encountered and all comments users have made on the usability of the PULS. Table 40 in Appendix 4 shows one example of usability test for one task including the detailed issues raised by users and the solutions made for those issues.

Cooperative evaluations for all tasks defined in the **prepare tasks** stage have been

performed like this table. After this, PULS has been improved and it is considered to be able to satisfy most users in terms of efficiency, effectiveness and ease of use in the representatives' viewpoint.

## 6.3 Evaluation

### 6.3.1 PULS IE system

As an Information Extraction system, the main task of PULS is to maximize the quality of extraction results, similarly to the general objective of other IE systems. The term *quality* here varies according to the extraction purpose and the evaluation metric. We continuously perform several types of evaluations on PULS IE systems ranging from the formal MUC-ACE style evaluations which measure the quality of the IE system results in terms of recall and precision for every slot in the template, to more coarse-grained evaluations which concentrate on a set of key slots in the template such as *disease name*, *country*, *date*.

**Setup** The first step is to **create testing key file and generate result file**. For each scenario and language set, i.e. medical-english, medical-russian, business-english, security-english, security-russian, we manually gather a testing corpus and each document in the corpus must have at least one appropriate event in the scenario and language. From the corpus, we create a testing key file that contains a number of templates with correct slots manually extracted from the documents in the corpus. Each template represents an event in a document (Table 1). We then use the corresponding PULS IE sub-system to process the same corpus, extract events from the documents as templates and append the templates into one result file. An example of templates extracted from disease outbreak document is shown below. This document contains three disease events and hence has three templates recorded.

```
<TEMPLATE-MED20001210_00.05.17_29312> :=
        DOC_NR:                 "MED20001210_00.05.17_29312"
        CONTENT:                <DISEASE_EVENT-MED20001210_00.05.17_29312-1>
                                <DISEASE_EVENT-MED20001210_00.05.17_29312-2>
                                <DISEASE_EVENT-MED20001210_00.05.17_29312-3>

<DISEASE_EVENT-MED20001210_00.05.17_29312-1> :=
        CASE_DESCRIPTOR:        57-year-old man   ##2154#2171#
        CASE_STATUS:            sick
        CASE_TOTAL:             27
        COMMENT_MATCHED_TEXT:   57-year-old man brought the number of confirmed cases in Minnesota to 27.
        COUNTRY:                USA/Minnesota   ##2213#2222#
        DISEASE_CANON:          Escherichia Coli
        DISEASE_NAME:           E. coli   ##2042#2049#
        LOCATION:               Minnesota   ##2213#2222#
        NORM_ETIME:             2000.12.07
```

```
        NORM_STIME:             2000.12.07
        TIME:                   Thursday [7 Dec 2000]   ##2379#2399#

<DISEASE_EVENT-MED20001210_00.05.17_29312-2> :=
        CASE_DESCRIPTOR:        3   ##2426#2427#
        CASE_STATUS:            sick
        CASE_TOTAL:             3
        COMMENT_MATCHED_TEXT:   One person in Iowa and 3 in Wisconsin also have been sickened.
        COUNTRY:                USA/Wisconsin   ##2431#2440#
        DISEASE_CANON:          Escherichia Coli
        DISEASE_NAME:           E. coli   ##2685#2692#
        LOCATION:               Wisconsin   ##2431#2440#
        NORM_ETIME:             2000.12.07
        NORM_STIME:             2000.12.07
        TIME:                   Thursday [7 Dec 2000]   ##2379#2399#

<DISEASE_EVENT-MED20001210_00.05.17_29312-3> :=
        CASE_DESCRIPTOR:        One person   ##2403#2413#
        CASE_STATUS:            sick
        CASE_TOTAL:             1
        COMMENT_MATCHED_TEXT:   One person in Iowa and 3 in Wisconsin also have been sickened.
        COUNTRY:                USA/Iowa   ##2417#2421#
        DISEASE_CANON:          Escherichia Coli
        DISEASE_NAME:           E. coli   ##2685#2692#
        LOCATION:               Iowa   ##2417#2421#
        NORM_ETIME:             2000.12.07
        NORM_STIME:             2000.12.07
        TIME:                   Thursday [7 Dec 2000]   ##2379#2399#
```

Suppose the corpus has 100 documents, both the key file and result file would have 100 such sets of templates. While the key file contains the right answers manually created by human for each document and each document has at least one template, the result file could have a different number of templates for the same document if the extraction result is wrong and sometimes it looks like the following if no template has been extracted by PULS IE system.

```
<TEMPLATE-MED20001210_00.05.17_29312> :=
        DOC_NR:                 "MED20001210_00.05.17_29312"
        CONTENT:
```

The next step is to **compare all the templates between the key file and result file**. We build a script for comparing them every time when something change in the PULS IE system (e.g., new patterns introduced). Since the number of extracted templates in a document could be different from the number in the answer key, the script is able to find the most probably matching templates within a document between key and result to compare. Further, the script is configurable to choose the slots for evaluation. For the formal MUC-ACE style evaluation, we compare every valuable slot in the template, i.e., all the slots as shown above. Alternatively, we focus on the key slots. For example, key slots for disease outbreak events include *case_ status*, *case_ total*, *country*, *disease_ canon*, *time*. Then, other slots are ignored and configured to be not scored in the script.

For each slot (e.g., *country*), precision, recall and f-measure (F) are calculated as described in Appendix 1a. Suppose the key file contains 200 templates, it will then

have 200 *N.key* country slots that need to be filled. If the PULS IE system fills 120 *N.correct[country]* slots and 120 *N.incorrect[country]* slots (including unfilled slots that should be filled), then,

$$Recall[country] = \frac{N.correct[country]}{N.key[country]} = \frac{120}{200} = 0.6$$

$$Precision[country] = \frac{N.correct[country]}{N.correct[country] + N.incorrect[country]} = \frac{120}{120 + 120} = 0.5$$

$$F[country] = \frac{2 \times Recall[country] \times Precision[country]}{Recall[country] + Precision[country]} = \frac{2 \times 0.6 \times 0.5}{0.6 + 0.5} = 0.545$$

we can see that, *N.correct[country]* + *N.incorrect[country]* can be larger than 200 since PULS IE system may extract some additional wrong events. The overall results are calculated using *N.correct* and *N.incorrect* for all the slots we want to evaluate like,

$$N.*[all] = N.*[country] + N.*[disease\_canon] + N.*[case\_status] + N.*[case\_total] + N.*[time] + ...$$

$$Recall[all] = \frac{N.correct[all]}{N.key[all]}$$

$$Precision[all] = \frac{N.correct[all]}{N.correct[all] + N.incorrect[all]}$$

$$F[all] = \frac{2 \times Recall[all] \times Precision[all]}{Recall[all] + Precision[all]}$$

*F[all]* is the final evaluation result of PULS IE system. we therefore try to balance the precision and recall to reach a higher F-measure score continuously.

**Improve existing IE systems** For existing IE systems, i.e., disease outbreak and business English IE sub-systems, we try to improve the quality of extraction results (NFR2) by fixing bad patterns and adding more appropriate patterns and inference rules. We then evaluate whether these patterns and inference rules can increase the recall while maintaining good precision. The Table 4 and Table 5 demonstrate the improvements made by the new appropriate patterns and others fixes in the disease outbreak and business IE sub-systems.

For disease outbreak scenario, we have manually created a key file containing 49 documents and 1275 slots as the right answers. These 35 documents are processed by our old disease outbreak IE system from 2010, 2011 and current improved system. The evaluation results are shown in Table 4.

For business scenario, the key file contains 116 documents, and 859 slots. The comparison of the evaluation results of old systems from 2010, 2011 and the current system is shown in Table 5.

From these two tables, the F-measure score of business IE system has increased by 17.3% while the score for disease outbreak IE system has increased by 5.8%. It indicates the focus has been largely placed on business scenario these years, and

| | **2010.08** | **2011.08** | **2012.08** |
|---|---|---|---|
| **Recall** | 0.67 | 0.66 | 0.68 |
| **Precision** | 0.7 | 0.79 | 0.78 |
| **F-measure** | 0.684 | 0.716 | 0.724 |
| **No. of documents**: 49<br>**No. of slots**: 1275 | | | |

Table 4: Improvements of PULS disease outbreak IE sub-system

| | **2010.08** | **2011.08** | **2012.08** |
|---|---|---|---|
| **Recall** | 0.54 | 0.62 | 0.66 |
| **Precision** | 0.54 | 0.55 | 0.62 |
| **F-measure** | 0.544 | 0.584 | 0.638 |
| **No. of documents**: 116<br>**No. of slots**: 859 | | | |

Table 5: Improvements of PULS business IE sub-system

also the higher the score, the more difficult to improve.

**New IE systems** For newly built IE systems, i.e., cross-border security English and Russian and disease outbreak Russian IE sub-systems, we apply the same evaluation metric as described above to evaluate the quality of the systems.

The results are changing all the time since we are continuously improving these new IE systems and adding more valuable examples in the keys. The current results for these new IE systems are shown in Table 6, Table 7 and Table 8. We can see that, the results are bad since they are in an early stage of development. For example, they extracts only a few slots from the document, whereas the keys contain all the slots. The disease outbreak Russian IE system currently only extracts *disease*, *country*, *case status* and *case total* currently, every missing slot therefore contributes a penalty score in the recall.

| **Date** | **Documents** | **Slots** | **Recall** | **Precision** | **F-measure** |
|---|---|---|---|---|---|
| 2012.09.14 | 36 | 247 | 0.15 | 0.26 | 0.193 |

Table 6: Evaluation result of PULS disease outbreak Russian IE sub-system

| Date | Documents | Slots | Recall | Precision | F-measure |
|---|---|---|---|---|---|
| 2012.09.14 | 49 | 645 | 0.48 | 0.43 | 0.456 |

Table 7: Evaluation result of PULS security English IE sub-system

| Date | Documents | Slots | Recall | Precision | F-measure |
|---|---|---|---|---|---|
| 2012.09.14 | 40 | 522 | 0.47 | 0.34 | 0.391 |

Table 8: Evaluation result of PULS security Russian IE sub-system

### 6.3.2 Relevance Classifier

Besides F-measure evaluation for slots as stated above, we propose a new metric for news surveillance system to help users to find their required events more quickly. The utility or relevance of an event, as described in [HVE11, HVD13], decided automatically by PULS IE system relevance classifier, is very useful for providing users better decision support. For example, by giving old events a lower relevance score and assigning recent outbreak events a higher score, people can more easily focus on the surveillance of current outbreaks by simply filtering out events with a lower relevance in PULS DS system.

For evaluating the performance of PULS relevance classifier, the user evaluated data is split into three classes; high-relevance (relevance scores 4 and 5), low relevance (relevance scores from 1 to 3) and irrelevant (relevance score of 0). The predictive power of our features is evaluated by using three different classifiers: Naive Bayes [JoL95], SVM [Pla99] and BayesNet [Bou04]. Evaluations are done using a 10-fold cross-validation. We evaluated the results using precision, recall, F-measure and accuracy.

**Business domain** In the business domain, we use 213 user-labeled events in 127 documents. Table 9 shows classification performance achieved on discourse, lexical and combined features. We currently utilize roughly 40 discourse-level features as described in Table 3. In Table 9, we report the system's performance on *all* events in our labeled corpus, as well as only on events that appear *first* within a document (which may contain more than one event). The first-event evaluation is interesting since we can view it as an additional document-level *text-filtering* task, where the relevance of the first event is used to define the relevance of the entire document.

We train two types of binary classifiers: the high-vs-low classifiers separate between

| | **Business Domain** | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *All events* | | | | | | *First events only* | | | | | |
| **High-vs-low** | **Lexical** | | **Discourse** | | **Combined** | | **Lexical** | | **Discourse** | | **Combined** | |
| SVM | 72.2 | (0.696) | 84.6 | (0.83) | **85.3** | **(0.833)** | 70.4 | (0.738) | 81.8 | (0.826) | 81.4 | (0.818) |
| Naive Bayes | 74.3 | (0.73) | 75.7 | (0.753) | 82.5 | (0.814) | 70.3 | (0.731) | 81.7 | (0.823) | 82.2 | (0.825) |
| Bayes Net | 75.3 | (0.73) | 84.2 | (0.823) | 84.5 | (0.823) | 71.6 | (0.718) | 81.5 | (0.822) | **82.8** | **(0.834)** |
| **Zero-vs-rest** | | | | | | | | | | | | |
| SVM | 81.0 | (0.894) | 84.8 | (0.916) | 82.6 | (0.904) | 84.0 | (0.912) | 84.4 | (0.914) | 84.7 | (0.916) |
| Naive Bayes | 84.8 | (0.915) | 83.0 | (0.906) | **85.5** | **(0.92)** | **89.2** | **(0.94)** | 83.6 | (0.91) | 86.2 | (0.924) |
| Bayes Net | 83.0 | (0.908) | 82.4 | (0.903) | 81.7 | (0.899) | 84.2 | (0.915) | 84.2 | (0.915) | 84.1 | (0.914) |

Table 9: Relevance classification results on business domain: accuracy and F-measure (in parentheses) for discourse features, lexical features, and combined features.

events labeled 4–5 and 0–3. The zero-vs-rest classifiers separate the zero-relevance (i.e., completely useless) events from the rest. In each case, the F-measure is calculated for predicting the higher-relevance class.

For each classifier, we show the performance using discourse features only, lexical features only, and the combined set of features. The classifiers are trained with feature selection using information gain. In the table, the bold score indicates the best score achieved for the given column.

**Medical domain** Table 10 shows the classification results using the same strategy as in business domain. In most cases, discourse features perform better than lexical features, and combining the discourse and lexical features improves the predictive performance over both discourse and lexical features alone. These classifications were obtained on approximately 900 events, in 530 documents.

| | **Medical Domain** | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *All events* | | | | | | *First events only* | | | | | |
| **High-vs-low** | **Lexical** | | **Discourse** | | **Combined** | | **Lexical** | | **Discourse** | | **Combined** | |
| SVM | 82.2 | (0.537) | **85.1** | (0.618) | 84.2 | (0.613) | 87.2 | (0.625) | 88.5 | (0.664) | **89.6** | (0.71) |
| Naive Bayes | 79.7 | (0.64) | 80.7 | (0.598) | 84.6 | **(0.702)** | 85.8 | (0.679) | 85.0 | (0.639) | 89.2 | **(0.728)** |
| Bayes Net | 80.6 | (0.558) | 79.1 | (0.615) | 79.5 | (0.64) | 82.6 | (0.529) | 82.0 | (0.612) | 82.5 | (0.619) |
| **Zero-vs-rest** | | | | | | | | | | | | |
| SVM | 83.9 | (0.907) | 84.8 | (0.913) | **85.9** | (0.917) | 80.6 | (0.888) | 81.6 | (0.895) | 83.0 | (0.897) |
| Naive Bayes | 85.3 | (0.915) | 84.1 | (0.908) | 85.7 | **(0.918)** | 82.7 | (0.898) | 82.5 | (0.895) | **83.8** | **(0.902)** |
| Bayes Net | 82.4 | (0.903) | 81.7 | (0.891) | 82.1 | (0.893) | 78.3 | (0.876) | 78.8 | (0.868) | 78.2 | (0.864) |

Table 10: Relevance classification results on medical domain

## 6.4 Chapter summary

In this chapter, internal testing, which aims to ensure all the new implemented functionalities work properly, has been presented first. External testing with the

attempt to ensure these new functionalities can satisfy the actual users' needs in both functional and non-functional point of view has been presented. Evaluations on the key functionalities are presented at last. After these testing and evaluations, PULS system is considered to be able to satisfy users' actual requirements.

# 7 Conclusion & Further work

This document has presented the whole work process followed to develop new features to improve PULS. In this chapter, overall process of this project will first be outlined. Then, suggestions for further work will be discussed and presented.

## 7.1 Conclusion

This project involves the development of new functionalities to improve PULS system to satisfy users' requirements. Specifically,

- In **PULS IE system**, besides improving the quality of extraction outcomes from English articles, a French pipeline and a Russian pipeline are required to be integrated into the system to handle articles in French and Russian. The cross-border security sub-system needs to be built to extract security incidents from related newspaper articles.
- In **PULS DS system**, the original table view, which directly presents the information from database table, should be improved for better decision support (e.g., be able to sort by any field, provide more advanced query, be able to save query, show more available fields, etc.). On current document page, a better way of verifying an event and assigning relevant level of an event are required. Further, more visualization interfaces including list view which groups similar events and display the list of groups, timeline view, graph view, map view etc., would provide much better decision support for users.

Since a clear plan has been made earlier and the appropriate process has been followed, all of these new functionalities as specified in Chapter 3 have been successfully implemented. After evaluation and testing, PULS is now working as expected to satisfy users' requirements. The whole process followed to develop PULS and what has been shown in this report are shown below.

- **Chapter 2 - Background**: the development of improvements starts with the investigation of the background of this project. In this chapter, background related to this project including *Information Extraction*, *text mining* and *Information Visualization* have been presented followed by programming languages.

- **Chapter 3 - Requirements**: after having investigated the background related to this project, we have created the requirement specification for improving PULS. In order to achieve this, several steps have been followed. In this chapter, old PULS system has been introduced first. A number of requirements for improving PULS from different sources have then been elicited and analyzed. Finally, all functional and non-functional requirements gathered have been assigned priorities to clearly present which requirements are keys and need to be firstly considered.

- **Chapter 4 - Design**: the most common and powerful approach called modelling has been chosen for the functional designs of PULS. In this chapter, new functionalities designed using UML have first been presented followed by user interface design and database design.

- **Chapter 5 - Implementation**: in this chapter, implementation technologies which have been chosen to implement PULS have first been investigated and introduced. All functionalities designed have been implemented and presented by first realizing the core functionalities and then implementing the optional functionalities. Finally, database implementation to accommodate new functionalities according to the database design has been presented.

- **Chapter 6 - Evaluation & Testing**: the purpose of evaluation and testing is to ensure that new PULS can work correctly and effectively. Evaluation and testing have been involved in all stages of development. For example, at the requirement stage, we have validated the requirements with actual users before creating the requirement specification; at the design stage, we have evaluated whether the design can correctly and completely accommodate the requirements identified in the requirement stage, etc. Besides these, a number of approaches have been used to test whether PULS can work correctly and effectively after the new functionalities have been implemented. In this chapter, internal testing which aims to ensure all the new functionalities can work properly by themselves have first been presented. External evaluation with the attempt to ensure these improvements can satisfy the actual users' needs have then been presented. Finally, evaluations of the main functionalities have been presented.

## 7.2 Suggestion for further work

Although all the new functionalities for PULS have been successfully developed, some of them are thought to be not good enough and can be improved. As an Information Extraction system, the main task of PULS is to maximize the quality of extraction results as same as the general objective of all other IE systems. In addition, how to better present the results to the end users is considered as another general objective of PULS in the long run.

Based on the investigation presented in the background chapter, requirements analysis as shown in the requirement chapter and users' feedbacks, some suggestions of further work for improving PULS are shown below.

- **Find more extraction patterns** using both *Knowledge Engineering Approach* and *Machine Learning Approach* described in Section 2.1.2 in all domains to increase the recall and evaluate these patterns to maintain reasonable precision.

- **Use knowledge base** that has been created and updated during the whole process of this project to improve both PULS IE system and provide more decision support for PULS DSS users. For example, as mentioned in Chapter 3, since we have collected a large number of sector tags that Esmerk manually created for each business news article in PULS knowledge base, we would be able to use such knowledge to automatically decide the business sector for any new un-read article in PULS IE system later.

- Since the relevance of event is considered as a very useful feature for users to filter out irrelevant events based on users' feedback, we need to **improve the quality of relevance classification system** for all domains.

- Besides improvements for PULS IE system stated above, a variety of other features could be provided for **better decision support** on PULS DS system. We may allow registered users for testing and evaluating our extraction results and decision support tools. We may also provide a number of questionnaires for users on DS systems in order to collect additional services they are interested in. Afterwards, these additional services can be provided if appropriate.

# References


5yrs11 WWW-5 years of infosthetics, 2011. http://moritz.stefaner.eu/projects/5yrs-infosthetics/. [2.8.2011]

ACE2000 ACE, Automatic Content Extraction 2000 Evaluation (ACE Phase 1 2000) Entity Detection and Tracking (EDT), 2000. http://www.itl.nist.gov/iad/mig/tests/ace/2000/. [21.6.2011]

ACE2001 ACE, Automatic Content Extraction 2001/2002 Evaluation (ACE Phase 2) Entity Detection and Tracking (EDT) + Relation Detection and Characterization (RDC), 2001. http://www.itl.nist.gov/iad/mig/tests/ace/2001/. [21.6.2011]

ACE2002 ACE, Automatic Content Extraction Summer 2002 Evaluation (ACE Phase 2b 2002), 2002. http://www.itl.nist.gov/iad/mig/tests/ace/2002/. [21.6.2011]

ACE2003 ACE, Automatic Content Extraction 2003 Evaluation (ACE03), 2003. http://www.itl.nist.gov/iad/mig/tests/ace/2003/. [21.6.2011]

ACE2004 ACE, Automatic Content Extraction 2004 Evaluation (ACE04), 2004. http://www.itl.nist.gov/iad/mig/tests/ace/2004/. [22.6.2011]

ACE2005 ACE, Automatic Content Extraction 2005 Evaluation (ACE05), 2005. http://www.itl.nist.gov/iad/mig/tests/ace/2005/. [22.6.2011]

ACE2007 ACE, Automatic Content Extraction & Entity Translation 2007 Evaluations (ACE07 & ET07), 2007. http://www.itl.nist.gov/iad/mig/tests/ace/2007/. [22.6.2011]

ACE2008 ACE, Automatic Content Extraction 2008 Evaluation (ACE08), 2008. http://www.itl.nist.gov/iad/mig/tests/ace/2008/. [22.6.2011]

ACE11 ACE, Automatic Content Extraction (ACE) Evaluation, 2011. http://www.itl.nist.gov/iad/mig/tests/ace/. [21.6.2011]

ADP13 Atkinson, M., Du, M., Piskorski, J., Tanev, H., Yangarber, R. and Zavarella, V., Techniques for multilingual security-related event extraction from online news. *In Computational Linguistics-Applications*. Springer Verlag, 2013.

AFH01 Anderson, R., Francis, B., Homer, A., Howard, R., Sussman, D. and Watson, K., *Professional ASP.NET*. Wrox, 2001.

AgG04 Agichtein, E. and Ganti, V., Mining reference tables for automatic text segmentation. *In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Seattle, USA, 2004.

AHB93 Appelt, E., D., Hobbs, R., J., Bear, J., Israel, J., D. and Tyson, M., Fastus: A finite-state processor for information extraction from real-world text. *IJCAI*, pages 1172–1178.

And92 Andersen, P. M., Hayes, P. J., Huettner, A. K., Schmandt, L. M., Nirenburg, I. B. and Weinstein, S. P., Automatic extraction of facts from press releases to generate news stories. *In Proceedings of the third conference on Applied natural language processing*, 1992, pages 170–177.

BaB97 Bagga, A. and Biermann, A. W., Analyzing the Complexity of a Domain With Respect To An Information Extraction Task. *Proceedings of the tenth International Conference on Research on Computational Linguistics (ROCLING X)*, 1997, pages 175–94.

BDS01 Borkar, R., V., Deshmukh, K. and Sarawagi, S., Automatic text segmentation for extracting structured records. *In Proceedings of ACM SIGMOD International Conference on Management of Data*, Santa Barabara, USA, 2001.

BMS97 Bikel, M., D., Miller, S., Schwartz, R. and Weischedel, R., Nymble: A high-performance learning name-finder. *In Proceedings of ANLP-97*, 1997, pages 194–201.

Bou04 Bouckaert, R., Bayesian network classifiers in Weka. 2004.

Bri96 Brigman, G., L., *Web site management excellence*. Que Corporation, 1996.

BSG98 Borthwick, A., Sterling, J., Agichtein, E. and Grishman, R., Exploiting diverse knowledge sources via maximum entropy in named entity recognition. *in Sixth Workshop on Very Large Corpora New Brunswick*, New Jersey, 1998.

Chi98 Chinchor, A., N., WWW-OVERVIEW OF MUC-7/MET-2, 1998. http://www-nlpir.nist.gov/related_projects/muc/proceedings/muc_7_proceedings/overview.html. [15.6.2011]

CLS11 WWW-CLSQL Overview, 2011. http://clsql.b9.com/. [28.9.2011]

CMB02 Cunningham, H., Maynard, D., Bontcheva, K. and Tablan, V., Gate: A framework and graphical development environment for robust NLP tools and applications. *In Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, 2002.

CMS99 Card, K., S., Mackinlay, D., J. and Shneiderman, B., *Readings in Information Visualization: Using Vision to Think.* Morgan Kaufmann Publishers, 1999.

DAR91 DARPA. *Proceedings of Proceedings of the Third Message Understanding Conference (MUC-3).* Morgan Kaufmann, 1991.

DAR92 DARPA. *Proceedings of the Fourth Message Understanding Conference (MUC-4).* Morgan Kaufmann, 1992.

DAR93 DARPA. *Proceedings of the Fifth Message Understanding Conference (MUC-5).* Morgan Kaufmann, 1993.

DAR95 DARPA. *Proceedings of the Sixth Message Understanding Conference (MUC-6).* Morgan Kaufmann, 1995.

DAR11 DARPA, WWW-Our Work, 2011. http://www.darpa.mil/our_work. [13.5.2011]

BMV12 Department of Computer Science, University of Helsinki, WWW-BMVis graph visualisation tool, 2012. http://www.cs.helsinki.fi/group/biomine. [13.2.2012]

Don03 Donaldson, I., Martin, J. D., de Bruijn, B., Wolting, C., Lay, V., Tuekam, B., Zhang, S., Baskin, B., Bader, G. D., Michalickova, K., Pawson, T. and Hogue, C. W. V., PreBIND and Textomy - mining the biomedical literature for protein-protein interactions using a support vector machine. *BMC Bioinformatics*, pages 4–11.

DVK11 Du, M., Von Etter, P., Kopotev, M., Novikov, M., Tarbeeva, N. and Yangarber, R., Building support tools for Russian-language information extraction. *Proceedings of the 14th international conference on Text, speech and dialogue*, Berlin, Heidelberg, 2011, Springer Verlag, pages 380–387.

Ear11 WWW-The Earth, Our Home in Space, 2011. `http://staff.on.br/~jlkm/astron2e/AT_MEDIA/CH07/CHAP07AT.HTM`. [25.7.2011]

EKM02 Evans, Kirk, A., Kamanna, A. and Muller, J., *XML and ASP.NET*. New Rider, 2002.

EMM11 EMM, WWW-Overview of Europe Media Monitor, 2011. `http://emm.newsbrief.eu/overview.html`. [12.3.2011]

ElN03 Elmasri, R. and Navathe, S., *Fundamentals of Database Systems*. Addison Wesley, 2003.

Esm11 Esmerk, WWW-The power to succeed, 2011. `http://www.esmerk.com/en/`. [12.2.2011]

Few06 Few, S., *Information Dashboard Design: The Effective Visual Communication of Data*. O'Reilly, 2006.

FHM06 Filatova, E., Hatzivassiloglou, V. and McKeown, K., Automatic creation of domain templates. *In Proceedings of the COLING/ACL on Main conference poster sessions (COLING-ACL '06)*, Association for Computational Linguistics, Stroudsburg, PA, USA, 2006, pages 207–214.

Fli11 FlipDog, WWW-Find Local Jobs & Employment Listings Job Search, 2011. `www.flipdog.com/`. [1.7.2011]

Gac04 Gachet, A., *Building Model-Driven Decision Support Systems with Dicodess*. VDF, 2004.

GAT11 GATE, WWW-GATE Developer, 2011. `http://gate.ac.uk/family/developer.html`. [12.7.2011]

GHY02 Grishman, R., Huttunen, S. and Yangarber, R., Real-Time Event Extraction for Infectious Disease Outbreaks. *In Proceedings of the 3rd*

*Annual Human Language Technology Conference HLT-2002*, San Diego, CA, 2002.

GaM89 Gazdar, G. and Mellish, C., *Natural Language Processing in Lisp*. Addison-Wesley, 1989.

GNU11 WWW-GNU Emacs, 2011. `http://www.gnu.org/software/emacs/`. [28.9.2011]

Goo12 WWW-Google Maps API, 2012. `http://code.google.com/apis/maps/index.html`. [13.2.2012]

GrS96 Grishman, R. and Sundheim, B., Message Understanding Conference-6: a brief history. *Proceedings of the 16th conference on Computational linguistics - Volume 1*, Copenhagen, Denmark, 1996, pages 466–471.

GaW98 Gaizauskas, R. and Wilks, Y., Information Extraction: Beyond Document Retrieval. *Computational Linguistics and Chinese Language Processing*, 3,2(1998), pages 17–60.

HCM04 Haag, S., Cummings, M. and McCubbrey, J., D., *Management Information Systems for the Information Age*. McGraw-Hill, 2004.

HFH09 Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I. H., The Weka data mining software: an update. *SIGKDD Explor. Newsl.*, pages 10–18.

Hig12 WWW-Highcharts, 2012. `http://www.highcharts.com/`. [13.2.2012]

Hob93 Hobbs, J. R., The Generic Information Extraction System. *In Proceedings of the Fifth Message Understanding Conference (MUC-5)*, 1993, pages 87–91.

HiS82 Hirschman, L. and Sager, N., Automatic Information Formatting of a Medical Sublanguage. *Sublanguage: Studies of Language in Restricted Semantic Domains*, pages 27–80.

HUN11 WWW-HUNCHENTOOT - The Common Lisp Web server formerly known as TBNL, 2011. `http://weitz.de/hunchentoot/`. [29.9.2011]

HVD13 Huttunen, S., Vihavainen, A., Du, M. and Yangarber, R., Predicting the relevance of event extraction for the end user. *Multi-source, Multilingual Information Extraction and Summarization*, pages 163–176.

HVE11 Huttunen, S., Vihavainen, A., Etter, von, P. and Yangarber, R., Relevance prediction in information extraction using discourse and lexical features. *Nordic Conference on Computational Linguistics*.

HYG02 Huttunen, S., Yangarber, R. and Grishman, R., Complexity of Event Structure in IE Scenarios. *In: Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, Taipei, August 2002.

IEB07 WWW-IEBuilder Toolkit InfoExtract, 2007. `http://www.infoextract.com/id30.html`. [26.6.2011]

JKR06 Jayram, T., Krishnamurthy, R., Raghavan, S., Vaithyanathan, S. and Zhu, H., Avatar information extraction system. *IEEE Data Engineering Bulletin*, 29, pages 40–48.

JoL95 John, G. H. and Langley, P., Estimating continuous distributions in bayesian classifiers. *In: Eleventh Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 1995.

Joa02 Joachims, T., *Learning to Classify Text Using Support Vector Machines: Methods, Theory, and Algorithms*. Springer, 2002.

Jon77 Jong, de, G., FRUMP...FRUMP...FRUMP... *SIGART Bull.*, ,61, pages 54–55.

Jon82 Jong, de, G., An Overview of the FRUMP System. *Strategies for Natural Language Processing*, pages 149–176.

Tag12 Jquery, WWW-TagCloud, 2012. `http://archive.plugins.jquery.com/project/TagCloud`. [29.2.2012]

JaR90 Jacobs, P. S. and Rau, L. F., SCISOR: extracting information from on-line news. *Communications of the ACM*, 33,11(1990), pages 88–97.

KiM95 Kim, J. and Moldovan, D., Acquisition of Linguistic Patterns for Knowledge-Based Information Extraction. *IEEE Transactiops on Knowledge and Data Engineering*, 7,5(1995), pages 713–724.

KlM02 Klein, D. and Manning, D., C., Conditional structure versus conditional estimation in NLP models. *in Workshop on Empirical Methods in Natural Language Processing (EMNLP)*, 2002.

KeS78 Keen, G. W., P. and Scott-Morton, S., M., *Decision Support Systems: An Organizational Perspective.* Addison-Wesley, 1978.

KrV05 Krallinger, M. and Valencia, A., Text-mining and information-retrieval services for molecular biology. *Genome Biology*, 6,224(2005), pages 1–2,5.

Wek12 learning group at University of Waikato, M., WWW-Weka 3: Data Mining Software in Java, 2012. `http://www.cs.waikato.ac.nz/ml/weka/`. [12.7.2012]

WMS93 Lehnert, W., McCarthy, J., Soderland, S., Riloff, E., Cardie, C., Peterson, J., Feng, F., Dolan, C. and Goldman, S., Umass/hughes: Description of the CIRCUS system used for tipster text. *In Proceedings of a Workshop on Held at Fredericksburg*, USA, 1993, pages 241–256.

Lyt86 Lytinen, S., Anatole Gershman ATRANS: Automatic processing of money transfer messages. *In Proceedings of the Fifth National Conference on Artificial Intelligence*, 1986, pages 1089–1093.

Mal02 Malouf, R., Markov models for language-independent named entity recognition. *In Proceedings of the Sixth Conference on Natural Language Learning (CoNLL-2002)*, 2002.

Mar99 Marakas, G., M., *Decision support systems in the twenty-first century.* Prentice Hall, 1999.

McC60 McCarthy, J., *Recursive Functions of Symbolic Expressions and Their Computation by Machine, Part I.* Massachusetts Institute of Technology, 1960.

McC96 McConnell, S., *Rapid Development.* Microsoft Press, 1996.

McC97 McConnell, S., *Software Project Survival Guide.* Microsoft Press, 1997.

McC58 McCarthy, J., WWW-LISP prehistory - Summer 1956 through Summer 1958, 2011. `http://www-formal.stanford.edu/jmc/history/lisp/node2.html`. [13.7.2011]

MFP00 McCallum, A., Freitag, D. and Pereira, F., Maximum entropy markov models for information extraction and segmentation. *In Proceedings of*

*the International Conference on Machine Learning (ICML-2000)*, CA, 2000, pages 591–598.

MaP98 Marsh, E. and Perzanowski, D., MUC-7 Evaluation Of IE Technology: Overview Of Results. *MUC*.

MTU01 Maynard, D., Tablan, V., Ursu, C., Cunningham, H. and Wilks, Y., Named entity recognition from diverse text types. *Recent Advances in Natural Language Processing 2001 Conference*, Bulgaria, 2001.

MWH93 Monk, A., Wright, P., Haber, J. and Davenport, L., *Improving your human-computer interface: a practical technique*. Prentice Hall, 1993.

MyS12 WWW-Why MySQL, 2012. `http://www.mysql.com/why-mysql/`. [13.2.2012]

OBJ06 Otasek, D., Brown, K. and Jurisica, I., Confirming protein-protein interactions by text mining. *the 6th SIAM Conference on Text Mining*, 2006.

Oja02 Ojala, M., WWW-WhizBang! Labs Closes Its Doors, 2002. `http://newsbreaks.infotoday.com/nbreader.asp?ArticleID=17168`. [23.6.2011]

Tho98 Powell, T. A., Jones, D. L. and Cutts, D. C., *Web Page Engineering: Beyond Web Page Design*. Prentice Hall, Upper Saddle River, 1998.

Pla99 Platt, J. C., Fast training of support vector machines using sequential minimal optimization. *In: Advances in kernel methods: support vector learning*. MIT Press, 1999.

Pro11 Promed-mail, WWW-About ProMED-mail, 2011. `http://www.promedmail.org/pls/apex/f?p=2400:1950:2676154989196101::NO:::`. [11.3.2011]

Pow02 Power, D., J., *Decision support systems: concepts and resources for managers*. Westport, Conn., Quorum Books, 2002.

PaR06 Patwardhan, S. and Riloff, E., Learning Domain-Specific Information Extraction Patterns from the Web. *ACL 2006 Workshop on Information Extraction Beyond the Document*, 2006.

PaR07 Patwardhan, S. and Riloff, E., Effective Information Extraction with Semantic Affinity Patterns and Relevant Regions. *EMNLP-CoNLL'07*, 2007, pages 717–727.

PhR07 Phillips, W. and Riloff, E., Exploiting Role-Identifying Nouns and Expressions for Information Extraction. *Proceedings of the 2007 Conference on Recent Advances in Natural Language Processing (RANLP-07)*, 2007.

QCS00 QCSA, WWW-Quintessential Careers Site Award: FlipDog.com, 2011. `http://www.quintcareers.com/Quintessential_Sites/FlipDog.html`. [1.7.2011]

RaA95 Ray, K. and Amy, S., *Interactivity by Design: Creating and Communicating with new Media.* Adobe Press, 1995.

Rat99 Ratnaparkhi, A., Learning to parse natural language with maximum entropy models. *Machine Learning*, 34.

Rij79 Rijsbergen, Van, C., *Information Retrieval.* Butterworths, 1979.

Ril93 Riloff, E., Automatically Constructing a Dictionary for Information Extraction Tasks. *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93)*, 1993, pages 811–816.

Ril96 Riloff, E., Automatically Generating Extraction Patterns from Untagged Text. *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, 1996, pages 1044–1049.

ScA77 Schank, R. and ABELSON, R., P., *Scripts, Plans, Goals, and Understanding.* Lawrence Erlbaum Associates, 1977.

Sag67 Sager, N., *Syntactic Analysis of Natural Language.* Academic Press, NY, 1967.

Sag75 Sager, N., Sublanguage Grammars in Science Information Processing. *Journal of the American Society for Information Science*, 26, pages 10–16.

Sag81 Sager, N., *Natural Language Information Processing: A Computer Grammar of English and Its Applications.* Addison-Wesley, 1981.

Sag11 Sager, N., WWW-Linguistic String Project (LSP), 2011. `http://www.cs.nyu.edu/sager/lspWWW.html`. [12.5.2011]

SBC11 WWW-Steel Bank Common Lisp, 2011. `http://www.sbcl.org/`. [28.9.2011]

SpC82 Sprague, H., R. and Carlson, D., E., *Building effective decision support systems*. Prentice-Hall, 1982.

Sch69 Schank, R., A conceptual dependency parser for natural language. *In Proceedings of the 1969 conference on Computational linguistics*, Sweden, 1969, pages 1–3.

Sch73 Schank, R., *Computer Models of Thought and Language*. Freeman, 1973.

Sch75 Schank, R., *Conceptual Information Processing*. North-Holland, 1975.

Sco71 Scott Morton, S., M., *Management Decision Support Systems: Computer-based Support for Decision Making*. Division of Research, Harvard University, 1971.

SDN07 Shen, W., Doan, A., Naughton, F., J. and Ramakrishnan, R., Declarative information extraction using datalog with embedded extraction predicates. *VLDB*, pages 1033–1044.

Sei05 Seibel, P., *Practical Common Lisp*. Apress, 2005.

SFA95 Soderland, S., Fisher, D., Aseltine, J. and Lehnert, W. G., CRYSTAL: Inducing a Conceptual Dictionary. *CoRR*.

SFL87 Sager, N., Friedman, C., Lyman, M., S. and members of the Linguistic String Project, *Medical Language Processing: Computer Management of Narrative Data*. Addison-Wesley, 1987.

SaG75 Sager, N. and Grishman, R., The Restriction Language for Computer Grammars of Natural Language. *Communications of the ACM*, 18, pages 390–400.

Shn96 Shneiderman, B., The eyes have it: a task by data type taxonomy for information visualizations. *Proceedings of the 1996 IEEE Symposium on Visual Languages*, 1996, pages 336–343.

SIM12 WWW-Semantic Interoperability of Metadata and Information in unLike Environments, 2012. `http://simile.mit.edu/`. [1.3.2012]

SLB94 Sager, N., Lyman, M., S., Bucknall, C., Nhan, N., T. and Tick, L., J., Natural Language Processing and the Representation of Clinical Data. *Journal of American Medical Informatics Society*, 1,2(1994), pages 142–160.

SLI11 WWW-SLIME: The Superior Lisp Interaction Mode for Emacs, 2011. `http://common-lisp.net/project/slime/`. [28.9.2011]

SMR99 Seymore, K., McCallum, A. and Rosenfeld, R., Learning Hidden Markov Model structure for information extraction. *in Papers from the AAAI-99 Workshop on Machine Learning for Information Extraction*, 1999, pages 37–42.

Sno1849 Snow, J., *On the Mode of Communication of Cholera*. J. Churchill, London, 1849.

Sno11 Snow, J., WWW-Snow's famous cholera map, 2011. `http://www.ncgia.ucsb.edu/pubs/snow/map.gif`. [22.7.2011]

Sod96 Soderland, S., G., Learning Text Analysis Rules for Domain-specific Natural Language Processing. Technical Report, Amherst, MA, USA, 1996.

Spr80 Sprague, H., R., A Framework for the Development of Decision Support Systems. *MIS Quarterly*, 4,4(1980), pages 1–26.

Ste02 Steele, G., WWW-Revenge of the Nerds, 2002. `http://www.paulgraham.com/icad.html`. [1.8.2011]

TuA97 Turban, E. and Aronson, J., *Decision support systems and intelligent systems*. Prentice Hall PTR Upper Saddle River, 1997.

TAL08 Turban, E., Aronson, E., J. and Liang, T.-P., *Decision Support Systems and Intelligent Systems*. Prentice Hall, 2008.

TeG03 Temkin, J., M. and Gilder, M., R., Extraction of protein interaction information from unstructured text using a context-free grammar. *Bioinformatics*, 19, pages 2046–2053.

Tim00 Timberlake, S., WWW-The Basics of Navigation, 2000. http://www.efuse.com/Design/navigation.html#SiteNavRoadmap. [13.5.2011]

TID11 WWW-TIDES Standard for the Annotation of Temporal Expressions, 2011. http://timex2.mitre.org/. [25.6.2011]

Tuf01 Tufte, E., *The Visual Display of Quantitative Information.* Graphics Press, 2001.

use11 useit.com, 2011. http://www.useit.com/. [15.8.2011]

War04 Ware, C., *Information visualization: Perception for design.* Morgan Kaufmann, 2004.

WDM09 WWW-Overview, 2009. http://wheredoesmymoneygo.org/dashboard/#year=2009&focus=TOTAL&view=uk-bubble-chart. [1.8.2011]

WDM11 WWW-Where does my money go?, 2011. http://wheredoesmymoneygo.org/. [1.8.2011]

WeR11 WebReference.com, 2011. http://www.webreference.com/. [13.8.2011]

WeT11 WebsiteTips.com, 2011. http://websitetips.com/. [13.8.2011]

Weg1912b Wegener, A., Die Entstehung der Kontinente. *Geologische Rundschau*, 3,4(1912), pages 276–292.

Weg1912a Wegener, A., Die Herausbildung der Grossformen der Erdrinde (Kontinente und Ozeane), auf geophysikalischer Grundlage. *Petermanns Geographische Mitteilungen in the annual meeting of the German Geological Society*, 1912, pages 185–195, 253–256, 305–309.

Weg1929 Wegener, A., *The Origin of Continents and Oceans (4 ed.).* Friedrich Vieweg & Sohn Akt. Ges, 1929.

Har11 Wikipedia, WWW-Harmonic mean, 2011. http://en.wikipedia.org/wiki/Harmonic_mean. [15.6.2011]

NLU11 Wikipedia, WWW-Natural language understanding, 2011. http://en.wikipedia.org/wiki/Natural_language_understanding. [13.6.2011]

Wil11 Williams, R., WWW-Good Design Features, 2011. `http://www.ratz.com/featuresgood.html`. [13.8.2011]

Yan06 Yangarber, R., Verification of Facts across Document Boundaries. *Proceedings of the International Workshop on Intelligent Information Access (IIIA-2006)*, Helsinki, Finland, August 2006.

# Appendix 1a. A brief history of IE systems

The history of Information Extraction can be traced back to 1960s in the early days of Natural Language Processing (NLP). Beginning in 1987, IE was greatly prompted by Defense Advanced Research Projects Agency (DARPA) [DAR11] which is an agency of the United States Department of Defense responsible for the development of new technology for use by the military. Through a series of Message Understanding Conferences (MUC) sponsored by DARPA, a number of research groups were stimulated to pursue IE. In general, research work on IE can be roughly divided into three categories mainly according to the time of occurrences: early work carried out before DARPA programme; work done in response to MUC; and a variety of recent research work after DARPA programme.

**Early work on template filling before DARPA** Since 1965, the Linguistic String Project by Prof. Sager and her colleagues at New York University has launched a number of researches in the IE field [Sag11]. The project started with studying the syntactic analysis of general natural language in English and shaping a comprehensive computer grammar towards studying a restricted semantic domain - medical sublanguage. The major results include: a parser and programming language for natural language grammars [Sag67, SaG75]; sublanguage methodology [Sag75]; and a medical language processor (LSP-MLP) to convert clinical documents to a semantic representation called *information format* of medical sublanguage [HiS82, SFL87, SLB94]. The so-called *information format*, which abstracted away from various natural language forms, is later well known as template as shown in Table 1. The work to extract *fact* from a document containing natural language text into the template was then called *template filling* and became the focus of IE research work. One interesting aspect to notice in LSP is that the template was not created by any expert in the medical sublanguage; rather, Sager and her colleagues tried to induce the information format using distributional analysis to discover word classes. This automatic way was replaced through 1980's by manual definition done by domain experts since it was too difficult at that time. Only recently, renewed interests for automatic creation of domain templates emerged [FHM06].

Another focus of early work was natural language understanding (NLU), which was a subtopic of NLP in artificial intelligence dealing with machine reading comprehension [NLU11]. A long term project dealing with NLU was carried out by Roger Schank and his colleagues since 1969 [Sch69, Sch73, Sch75]. Finding certain stereo-

typical patterns that the event of stories followed is the main idea of the work. By using the patterns which Schank referred to as scripts, computer would extract any instance of event that the scripts might derive in a newspaper article. The first IE system using this approach was designed and built by one of Schank's students, Gerald De Jong. The system called *FRUMP* [Jon77, Jon82] used a simplified version of Schank's detailed scripts [ScA77] to skim newspaper stories and understand the most important points in a newspaper article and generate summary of the story. FRUMP relied on an alternation of two modules, *predictor* and *substantiator*. Relying on predictions from the scripts, *predictor* used the top-down and expectation-driven processing while *substantiator* applied the bottom-up and data-driven approach based on the input of the text. The alternation of script-based or data-driven approach has been used by many of later IE systems.

Following the two projects which shaped the idea of template filling and alternation of script-based or data-driven approach described above, the first commercial IE system was developed in 1980's called *Automatic processing of money transfer messages (ATRANS)* [Lyt86]. ATRANS, which combined the idea of both prior projects, adopted the Schank's script-driven approach to process the text, identify actors including *originating customer*, *originating bank*, *receiving bank*, etc., and finally automatically fill the template (format understandable by a bank's automatic payment system) that needed these roles. After being verified by human, the system initiated automatic money transfer. The system successfully released human from reading unformatted natural language money transfer messages and encoding such messages into system format which was relatively slow and expensive. It also illustrated the solutions to a number of problems encountered when applying an academic theory to a real-world problem [Lyt86]. Other notable systems initiated in this period included *JASPER* [And92], *SCISOR* [JaR90], etc.

**Work in response to MUC** Starting from 1987, IE was spurred by a number of Message Understanding Conferences which were sponsored by DARPA and organized by the US Naval Ocean System Center. The first conference, MUC-1 was introduced by DARPA in order to understand and compare their IE systems' behavior better. Six more conferences followed until 1998 and these MUCs have greatly driven IE forward. The term "message understanding" was replaced by more descriptively accurate - "Information Extraction" during this period. In response to DARPA's requirements, the objective of the conferences was to establish a quantitative regime for the evaluation of the IE system instead of the usual ad hoc way,

that trained and tested the system using the same data. They built answer keys for each document in test set and compared the extraction results with the answer keys to evaluate the extraction results. A brief summary of these MUCs can be found in Table 11 and Table 12 [DAR91, DAR92, DAR93, DAR95, GrS96, Chi98, MaP98].

| **Name** | **Year** | **Focus area** | **Participants** | **Train set** | **Test set** |
|---|---|---|---|---|---|
| MUC-1 | 1987 | Naval operations messages | 6 | - | - |
| MUC-2 | 1989 | Naval operations messages | 8 | 105 | 20+5 |
| MUC-3 | 1991 | Terrorism in Latin American countries | 15 | 1,300 | 300 |
| MUC-4 | 1992 | Terrorism in Latin American countries | 17 | - | 200 |
| MUC-5 | 1993 | Joint ventures and microelectronics domain | 17 | 2300 | 200+286 |
| MUC-6 | 1995 | News articles on management changes | - | 400 | 260 |
| MUC-7 | 1998 | Satellite launch reports | - | 200 | 200 |

Table 11: summary description of MUCs

| **Evaluation tasks** | **Named Entity** | **Corefer.** | **Templ. Element** | **Templ. Relation** | **Scenario Templ.** | **Multi Lingual** |
|---|---|---|---|---|---|---|
| MUC-1-4 | - | - | - | - | YES | - |
| MUC-5 | - | - | - | - | YES | YES |
| MUC-6 | YES | YES | YES | - | YES | - |
| MUC-7 | YES | YES | YES | YES | YES | - |

Table 12: Tasks evaluation of MUCs

According to Table 11 and Table 12, there was a notable increase in task complexity on several measures including *corpus complexity* (e.g. number of articles, average sentence length, vocabulary, etc), *template characteristics* (e.g. number of slots, number of object types, etc.), *evaluations* (e.g. evaluation metrics, size of test cor-

pus, etc.) and *task difficulty* (e.g. number of different tasks, template fill definitions, etc.) from MUC-1 to MUC-7. However, the average performance of those participating systems had not dropped (see Table 13). Considering the backdrop of the increasing task complexity, the work that had been made in response to these MUCs could be considered as genuine progress in developing IE technology.

| | **Named Entity** | **Corefer.** | **Templ. Element** | **Templ. Relation** | **Scenario Templ.** | **Multi Lingual** |
|---|---|---|---|---|---|---|
| MUC-3 | - | - | - | - | R<50%; P<70% | - |
| MUC-4 | - | - | - | - | F<56% | - |
| MUC-5 | - | - | - | - | EJV: F<53%; EME: F<50% | JJV: F<64%; JME: F<57% |
| MUC-6 | F<97% | R<63%; P<72% | F<80% | - | F<57% | - |
| MUC-7 | F<94% | F<62% | F<87% | F<76% | F<51% | - |

Table 13: Maximum results by task of MUCs

Besides the boost of IE technology brought by MUCs, one of the most important contributions of these MUCs is the development of evaluation metrics of IE system. The evaluation metrics have being evolving along with MUCs. Primary measure started from the standard IR metrics of recall and precision. During MUC-2, the detailed definition of recall and precision was worked out for IE. Suppose the answer key has N.key slots that need to be filled, and the system filled N.correct slots and N.incorrect slots (including unfilled slots that should be filled), then,

$$Recall = \frac{N.correct}{N.key}$$

$$Precision = \frac{N.correct}{N.correct + N.incorrect}$$

Recall and Precision were the primary metrics for MUC-3 and MUC-4. Additional measures have been also introduced as secondary measures. For example, it was suggested that the slot fills could be correct, partially correct, incorrect, missing or spurious (been filled when it should not be). These extra categories allowed

the introduction of measures like over-generation, under-generation and substitution [DAR91, DAR92]. During MUC-4, Rijsbergen's combined measure of recall and precision called *F-measure* was also introduced for IE [Rij79]. Another primary metric called *error per response fill*, which measured the fraction of a system's response that was *wrong*, was introduced for MUC-5 [DAR93]. However, the official evaluation metric reverted to precision, recall and F-measure in MUC-6 [DAR95] and MUC-7 [Chi98] because participants generally preferred precision and recall. Among all the evaluation metrics introduced during MUCs, F-measure, the weighted harmonic mean [Har11] of recall and precision as shown below, was later considered as the most frequently used primary measure of IE systems until now.

$$F = \frac{2 \times Recall \times Precision}{Recall + Precision}$$

**Recent work after MUC** After MUC-7, IE research has been stimulated by the *Automatic Content Extraction* (ACE) evaluations, whose objective is to develop automatic content extraction technology to support automatic processing of human natural language in free-text form from a variety of sources (e.g. newswire, broadcast conversation, etc.) [ACE11]. A summary of ACE evaluations can be found in Table 7.2.

From 2000, the ACE started with the *Entity Detection and Tracking* (EDT, later called *Entity Detection and Recognition, EDR*) which required that the selected types of entities in the text and their selected attributes be detected and merged into an unified representation format which included attributes and mentions of the entity [ACE2000]. Besides the EDR, In 2001, ACE required that the relations between the identified set of entities (RDR) could be determined [ACE2001]. Multilingual evaluation was introduced in 2003 when the scope of the data was expanded to include text from two new language (*Chinese* and *Arabic*) in additional to *English* [ACE2003]. ACE *Event Detection and Recognition* task (VDR) and *Timex2 Detection and Recognition* task [TID11] (TDR, a.k.a."TERN", for Time Expression Recognition and Normalization) was required for evaluation in 2004 [ACE2004]. More evaluation tasks had been introduced from 2005 to 2007, the sources of the corpus had also expanded from only newswire and broadcast news to broadcast conversations, web-log, usenet and conversational telephone speech [ACE2005, ACE2007]. ACE08, the most recent ACE evaluation, took place in May 2008 which focused only on two tasks (EDR and RDR) for English and Arabic articles locally (within-document) and globally(cross-document) [ACE2008].

Before each evaluation meeting, a clear evaluation plan which contained the task definition, detailed evaluation metric and tool for each task and the corpus information was posted on ACE official website [ACE11] in order to guide the evaluation process. The evaluation results were also posted after the evaluation events to keep the competitive development of IE technologies among different IE systems to improve performance on the tasks.

| Year | EDR | RDR | VDR | TERN | EMD[a] | RMD[b] | VMD[c] | Languages |
|---|---|---|---|---|---|---|---|---|
| 2000 | Y | N | N | N | N | N | N | EN[d] |
| 2001 | Y | Y | N | N | N | N | N | EN |
| 2002 | Y | Y | N | N | N | N | N | EN |
| 2003 | Y | Y | N | N | N | N | N | EN,CH[e],AR[f] |
| 2004 | Y | Y | Y | Y | N | N | N | EN,CH,AR |
| 2005 | Y | Y | Y | Y | N | Y | Y | EN,CH,AR |
| 2007 | Y | Y | Y | Y | N | Y | Y | EN,CH |
| 2008 | Y | Y | Y | Y | N | Y | Y | EN,AR |

[a]EMD: Entity Mention Detection
[b]RMD: Relation Mention Detection
[c]VMD: Event Mention Detection
[d]EN: English
[e]CH: Chinese
[f]AR: Arabic

Table 14: A chronological summary of Automatic Content Extraction evaluations

In general, ACE series of evaluations had greatly prompted the development of human language understanding technologies from the basic tasks like EDT and RDR to more advanced tasks during the last decades. Along with these ACE evaluations, a number of IE systems have been developed for different purposes in a variety of domains. Here are some examples:

- In 2000, WhizBang! Labs launched its first product, a job search website called *FlipDog* [Fli11]. By using the IE technology from WhizBang! Labs [Oja02] that spidered corporate websites for job postings, automatically extracted the data, categorized the jobs extracted, and added them to FlipDog's database, FlipDog claimed that it collected jobs that would not be seen from any other job site and it had more jobs listed than Monster and Hotjobs combined whose job openings were from manual posts. FlipDog won Quintessential Careers Site Award [QCS00] and were acquired by Monster in 2001.

- In New York University, Grishman, Huttunen and Yangarber developed a real-time extraction system called *Proteus BIO* [GHY02] that gathered daily news from multiple news sources, extracted them and automatically updated the extracted information to a database. The information extracted was infectious disease outbreaks containing disease name, location, date, and other fields. A Web-based browser then allowed users to query any outbreaks by these fields. The system provided an alternative way for retrieving documents that contained the actual outbreak events quickly and could potentially save a lot of time for manually going through those possible documents that contained only the disease name keywords. Patwardhan and Riloff [PaR07], Phillips and Riloff [PhR07] also worked on IE system in disease outbreak domain.

- In biomedical and molecular biology domain, there were a large number of IE applications developed in recent years for different purposes and usages, While some applications covered the full process of text mining from filtering related documents to mining the biomedical interactions and relations from the documents, some of them only provided useful and effective tools for one or more stages of the full process [KrV05]. Protein-protein interactions automatically extracted from the scientific manuscripts were the most interesting focus in this domain [Don03, TeG03, OBJ06].

Besides the development of IE systems that were used for a specific purpose, another recent trend of IE research is to provide general tools for building IE systems. For example, *IEBuilder Toolkit*, developed by InfoExtract, is an Information Extraction systems that aims in providing tools to build high-performance, multilingual, adaptive, and platform-independent natural language processing [IEB07]. *GATE Developer*, developed by University of Sheffield, is another example of module based IE development environment [GAT11]. It provides a number of IE modules and rich set of graphical interactive tools for creating, measuring and managing the IE applications.

# Appendix 1b. History of Information Visualization

Information Visualization is not a new concept for the purpose of displaying IE outcomes. The oldest history of IV can be traced back to two centuries ago for visualizing statistical data [Tuf01] or even earlier for map-making. Let's see two interesting examples in the history that demonstrate the power of visualization in penetrating the surface and revealing the hidden message:

- In 1849, Dr. John Snow published a pamphlet *On the Mode of Communication of Cholera* in which he proposed the idea that the *Cholera Poison* reproduced in the human body and was spread through contaminated food or water [Sno1849]. His innovative theory, which argued against commonly accepted idea that Cholera, like all diseases, was transmitted through inhalation of contaminated vapors, was not approved with the technology and knowledge at that time. Until 1854, when a serious Cholera outbreak struck England again in London when Snow was able to prove his theory. In investigating the epidemic, Snow began to plot the location of deaths related to the outbreak and the pump distribution on a map (Figure 21). Through analyzing the combined mapping, Snow had found it was surprisingly obvious that the deaths were around a pump in Broad Street. The testing of the pump proved that the terrible decision of dumping wastes into the Thames river made by the government in London was the cause of the outbreak. Snow's theory is approved by this case and his innovative study of the mapping demonstrates the powerful value of spatial analysis in understanding and resolving a social problem.

- Alfred Wegener, a German meteorologist, is famous for his theory of *continental drift*. The first thought of this idea dramatically happened when he was sick in bed in 1910. Wegener was staring at the world map on the wall, and the underlying message of the map suddenly started emerging, *"the different large landmasses of the earth fit so closely like a jigsaw when moving them together"* [Weg1929] (Figure 22). From 1912, Wegener started publicly advocating the theory of *continental drift* by presenting a number of circumstantial evidence [Weg1912a, Weg1912b, Weg1929]. Although he did not come up with a convincing mechanism before he died in 1930 and his hypothesis was rarely accepted before 1960s, numerous developments began providing supporting evidence since 1960s and Alfred Wegener was then quickly recognized as the

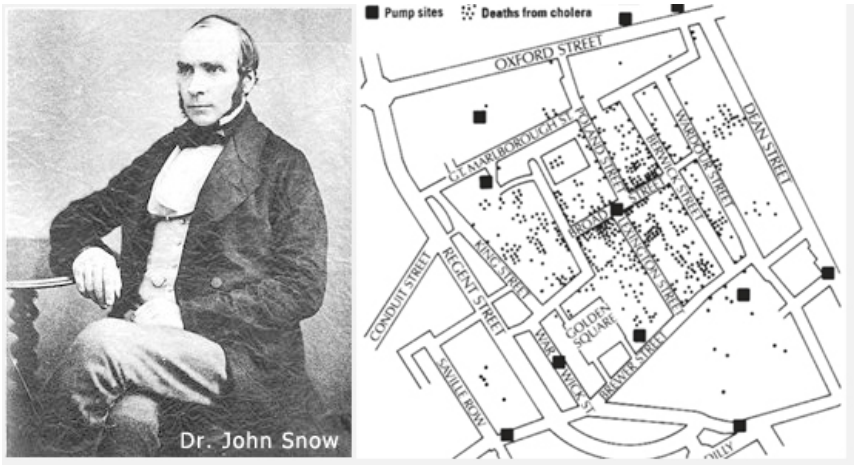


Figure 21: Pump sites and deaths from Cholera distribution map [Sno11]

founding father of one major scientific revolution in 20th century.

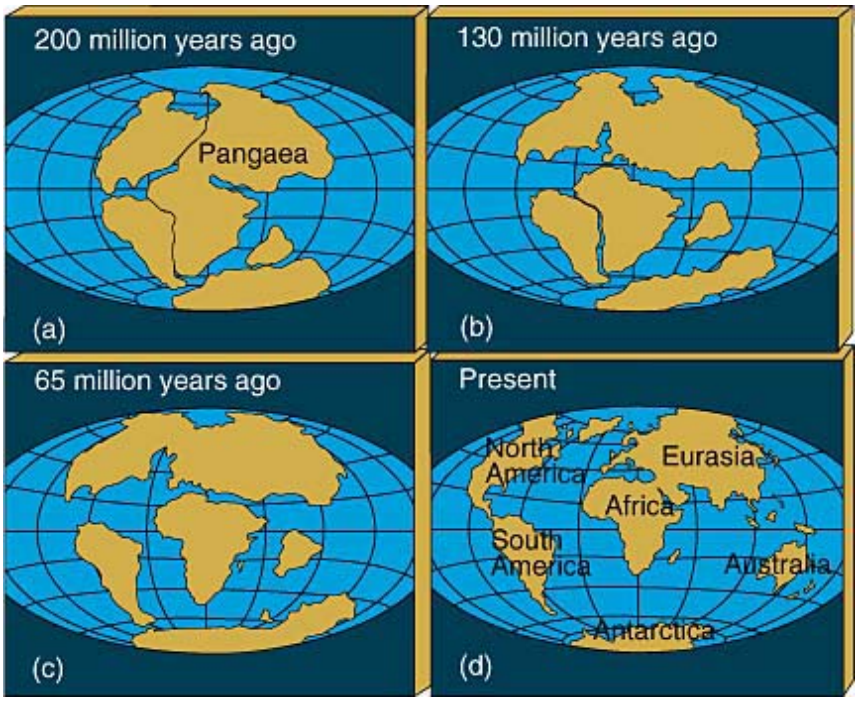


Figure 22: *Continental drift* theory [Ear11]

Since 1990s, the newly coming graphical user interface enables people to directly interact with the visualized information and perform *dynamic queries* on it. This is one of the most important features of computer-based IV. By using this feature, user is able to for example focus on the important part and to dynamically explore the detailed information of that part which is usually called *drill-down* [War04]. The famous *visual information seeking mantra* summaries this idea as *"Overview first, zoom and filter, then details-on-demand"* [Shn96]. Another similar study called *Three viewing depths* for a statistical graphic suggests almost the same thing. In most visual problem-solving tasks, both *large-scale overview* which facilitates the comparison of data points and *detailed view* which enables users to lookup and analyze data set [War04] together help users to effectively discover the valuable information inside the abstract data. The development of these new interactive visualization methods makes IV become a separate discipline. Term "Information Visualization" is created by Stuart Card, Jock Mackinlay and George Robertson in 1989 and they defined IV as the *"use of interactive visual representations of abstract, nonphysically based data to amplify cognition"* in 1999 [CMS99].

# Appendix 1c. Example figures of IV

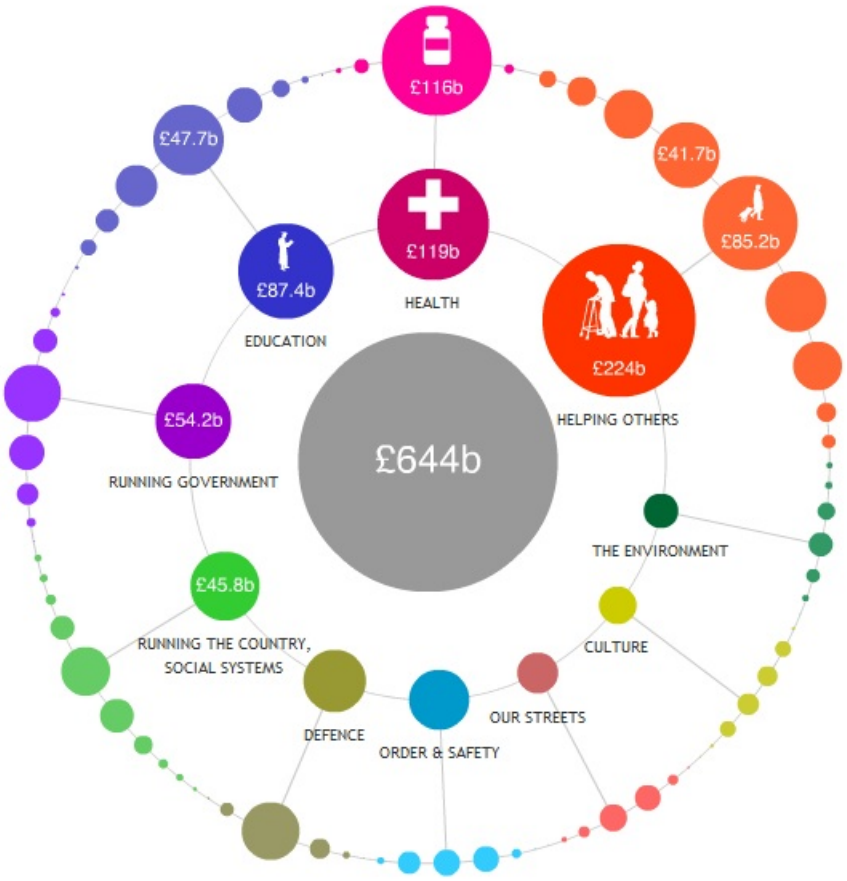


Figure 23: Where does my money go? - 2009 Overview [WDM09]


Figure 24: 5 Years Infosthetics [5yrs11]

# Appendix 1d. Development History of DSS

The concept of *Decision Support System* emerged in 1970s [Sco71, KeS78]. In 1980, Sprague proposed that a DSS was formed by three components: dialogue component, data component and model component [Spr80], such definition of the general structure of DSS has greatly promoted the development of DSS.

Starting from late 1980s and early 1990s, DSS and Expert System (ES) were combined to form the Intelligent Decision Support System (IDSS) which has the advantages of both DSS and ES. While keeping the characteristic of ES which solves qualitative analysis problem by inference using knowledge, IDSS also has the ability of DSS to solve the quantitative analysis problem by computational modelling. IDSS, which has achieved the combination of qualitative and quantitative analysis and has greatly promoted the development of the computer-based problem-solving abilities, is considered as a new development stage of DSS.

In the mid-1990s, new technologies including Data Warehouse (DW), On-Line Analysis Processing (OLAP) and Data Mining (DM) have gradually shaped a new concept for DSS while the IDSS was then classified as the traditional DSS. New DSS is characterized by acquiring information and knowledge from data which is completely different from the traditional DSS that provides decision support by utilizing model and knowledge-base. Traditional DSS and new DSS provide two different decision support mechanisms. They cannot substitute each other, but should be combined with each other.

A system combining both traditional and new DSS is considered as a more advanced form of DSS, called *Synthetic Decision Support System (SDSS)*. A SDSS, which takes the advantages of both traditional and new DSS in assisting decision-making, would achieve more effective decision support and is the future development direction of DSS.

In addition, along with the development of the Internet and network environment, DSS will be constructed in a new structure. The resources of DSS, such as data, model, knowledge etc., will be utilized as shared resources on the network server to provide concurrent shared services. Both management principle of the knowledge based economy - Knowledge Management (KM) and the next-generation Internet technology - Grid Computing (GC) are related to DSS. While KM system emphasizes on knowledge sharing and GC emphasizes on resource sharing, Web-base DSS uses the shared decision-making resources to provide decision-making assistance. A

new Data Warehouse based DSS is the application of KM technologies. In the network environment, DSS is constructed based on GC technologies, and is fully using shared decision-making resources on the grid, to achieve change-on-demand decision support. Such Web-base DSS opens up a new path for the development of DSS.

# Appendix 1e. Introduction to XML

One of the most far-reaching new ideas in computing is the evolution of Extensible Markup Language (XML) which *"have matured into standards that are being adopted by almost every sectors of the industry"* [AFH01].

What is XML? The book called *XML and ASP.NET* may present us a very simple but clear answer: *"XML is just data. XML is not a message protocol; it is not a wire transfer specification. XML is just data and how data is represented in a unified way."* [EKM02].

**Why XML?** Data exchange is a very important job in the network environment. However, there is always a big problem of data exchange among computers simply because there are a variety of document formats. Different software companies have different document formats of their own. In fact, there is no document format which can exchange data with all other document formats easily except those document formats belonging to the same type of application. For example: database file of Access and SQL Server can exchange data without much problem. Currently, in the world of networks, every website can be seen as a data source and they all have different structure of data. If we want to mine all the data on the network, we should first solve the problem of the integration of those data with different structures which are known as semi-structured data.

*Semi-structured* is a major characteristic of data on the Web. According to this characteristic, we need to find a semi-structured data model to clearly describe data on the Web. XML provides a good solution. A simple example of XML document is shown below:

```
<?xml version="1.0" encoding="utf-8"?>
<resume>
    <generalInfo>
        <Name>Mian Du</Name>
        <Title>MR</Title>
    </generalInfo>
    <workingInfo>
        <job>
            <company>Abc</company>
            <position>computer consultant</position>
        </job>
        <job>
            <company>Xyz</company>
            <position>general manager</position>
        </job>
    </workingInfo>
</resume>
```

As a semi-structured data model, XML is in accordance with those existing Web

applications and can easily achieve the data sharing and exchanging on the Web.

**Appearance and development of XML** XML is designed by W3C and the version 1.0 was released on Feb 1998. It is a very important embranchment of Standardized Generalized Markup Language (SGML) which inherits the advantages of SGML and omits complexities of SGML. In general, XML is a kind of Meta Markup Language which can provide format to describe structured data. Specifically, XML is a language designed to describe data similarly to HTML and it is able to support the disadvantages of TAG in HTML; XML is composed of several rules, these rules can be used to create Markup Language; and XML also use a simple application called *analyzing application* to process all newly developed Markup Language. Same as HTML, which has provided the first Internet users a way of showing internet documents, XML has created a global language which can be read and written by anybody. XML has solved a problem which can not be solved by HTML, i.e., one cannot find required information easily among a variety of useful information.

The TAG in XML has not been pre-defined. Users should define required tags of their own. XML does not try to organize contents; instead, it tries to describe them. In the next section, some basic characteristics of XML document will be presented.

**Characteristics of XML document** Usually, we should pay attention to two basic characteristics of XML documents, *Well-Formedness* and *Validation*.

1. **Well-Formedness**: XML documents belong to well-Formed documents. In contrast to HTML documents, we should always include end tags in XML documents. For example: *<company>Xyz</company>* in previous section. We should have an end tag *</company>* for the begin tag *<company>*; or for some tags which do not have single end tags, we could write like *<mailaddress type= "from" />*.

2. **Validation**: due to that all tags in XML are defined by users, XML documents do not have default tags and structures. We need to use Document Type Definition (DTD) or XML Schema to check whether those definitions of tags in XML documents are in accordance with the grammars of XML.

The main usages of XML are shown below:

- XML can store information and provide them to HTML in order to dynamically generate HTML web pages.

- XML can be a data format for exchanging data: you only need to convert a variety of data formats into XML documents in order to exchange data among different systems, especially on the Internet.
- XML can be a data format to store data: XML can be one data format for applications to store data. As XML both support database files and document files, it is very easy for all application systems to store information using XML format by developing some application codes.

Many application systems of enterprises support XML as it provides a data exchanging format which is considered as the best solution for data exchange among internal and external application systems of enterprises. How XML is used in PULS system is introduced in Chapter 5.

# Appendix 1f. What made Lisp different?

In the development history of programming languages, Lisp represents an important programming mindset. Lisp, which is based on mathematics and logics, has been directly effecting the evolution of modern high-level programming language.

Nine new ideas were created when Lisp was born more than 50 years ago. While some of them are commonly accepted today, some others just appear in other high-level languages and two kinds are still unique so far. The nine ideas, in the order of their adoption by the mainstream, are [Ste02],

1. **Conditionals**: (i.e. "if-then-else structure").
2. **Function is a data type**: in Lisp, like integers or strings, functions is one type of data. They have their own literal representations, can be stored in variables or be passed as parameters, etc.
3. **Recursion**: Lisp is the first high-level language that support recursive functions.
4. **Dynamic typing**: In Lisp, each variable is actually a pointer. The value it is pointing to has a type but the variable itself has not. Assigning or binding the value to a variable means copying the pointer rather than copying the value.
5. **Garbage collection**
6. **Programs are formed by expressions**: Lisp program is a collection of expression blocks in a tree structure. Each of the expression blocks returns a value. This is very different with Fortran and most succeeding languages which separate expressions and statements. Distinguishing between expressions and statements in Fortran is quite natural since you could not nest statements. And so while using mathematical formulas to compute a value, you would have to use expression to return this value and there is no other way to be used to handle it. Later, this limitation went away with the arrival of block-structured language. It was however too late by then since the distinction between expressions and statements was entrenched and it spread from Fortran to Algol followed by the subsequent spread into both their descendants.
7. **Symbol type**: symbols are actually pointers to strings stored in a hash table. In order to check if two strings are equal, Lisp simply check if they have the same pointer and does not need to compare their characters one by one.

8. **A notation for code using trees of constants and symbols**

9. **The entire language is available at any time**: Lisp does not really distinguish the read-time, compile-time and runtime. For example, this would allow you to:

    - compile or run code at read-time which allow users to reprogram Lisp's syntax;
    - run code at compile-time which is the basis of Lisp macros;
    - compile at runtime which is the basis of Lisp's usage as an extension language in programs like Emacs;
    - read at runtime which enables programs to communicate using s-expressions (SEXP).

When Lisp was born, these ideas were far removed from other programming languages which were designed according to the hardware in the late 1950s. As time goes by, the continuous upgrading of the popular languages had gradually evolved them toward Lisp. Ideas 1 to 5 are widely accepted nowadays; Idea 6 began to appear in the mainstream languages recently; Number 7 has been achieved in Python.

Ideas 8 and 9 were not from McCarthy's original vision and were added by his student Steve Russell. They make Lisp look strange but also unique. The strange look of Lisp is not because of its strange syntax but because it has no syntax. Lisp uses the form of parse trees which consist of its basic structure - lists to express the program directly, while parse trees are normally generated behind the scenes when other languages are parsed. Expressing the language using its own data structure has been proved to be a very powerful feature. Ideas 8 and 9 means than you may write programs to write programs. This may sound bizarre, but it is used every day in Lisp called *Macro*. The meaning of term *Macro* in Lisp is not the same in other languages. Lisp Macro means everything you may do in programming from an abbreviation of expressions to a new language compiler and it is still unique among all other languages.

Programming Language is not just a technology, it can also be a habit of thinking. Different languages affect our brain thinking in different ways. Today, Lisp is more recognized as a powerful programming idea and its various dialects based on its characteristics and ideas are working in their respective fields. *"If you think the greatest pleasure in programming comes from getting a lot done with code that simply*

*and clearly expresses your intention, then programming in Common Lisp is likely to be about the most fun you can have with a computer. You'll get more done, faster, using it than you would using pretty much any other language."* [Sei05]

# Appendix 2. PULS detailed requirements specifications

| Requirement ID | Description |
|---|---|
| IE-1 | Develop and integrate an IE sub-system for cross-border security domain into PULS IE system |
| IE-2 | Develop and integrate a French pipeline for disease outbreak domain into PULS IE system |
| IE-3 | Develop and integrate a Russian pipeline for cross-border security domain into PULS IE system |
| IE-4 | Integrate the new business source into PULS IE system |
| IE-5 | Extend the French pipeline and Russian pipeline to other domains |

Table 15: Requirements of PULS IE system elicited by PULS changes

| Requirement ID | Description |
|---|---|
| UI-1 | Provide user interface for cross-border security domain |
| UI-2 | Change encoding system to UTF-8 in user interface for displaying French and Russian characters |
| UI-3 | Provide both integrated and separate table views for different event types of business and security domain |
| UI-4 | Provide new document page with different slots for different event types of business and security |

Table 16: Requirements of PULS user interface elicited by PULS changes

| Requirement ID | Description |
|---|---|
| DB-1 | Create database for cross-border security domain |
| DB-2 | Change encoding system to UTF-8 in database for storing French and Russian characters |
| DB-3 | Merge old business databases (one database for one business event type) into one database to be able to better support integrated view and easily accomadate new event type |

Table 17: Requirements of PULS database elicited by PULS changes

| Requirement ID | Description |
|---|---|
| IE-6 | Increase recall of PULS IE system results for all domains |
| IE-7 | Increase precision of PULS IE system results for all domains |
| IE-8 | Balance the recall and precision to maximize the F-measure score for all domain |
| IE-9 | Create more suitable test sets for all domains for better evaluation purpose |

Table 18: Requirements of PULS IE system elicited from literature resources

| Requirement ID | Description |
|---|---|
| UI-5 | Make the website's purpose clear by explaining who you are and what you do. |
| UI-6 | Help users find what they need. |
| UI-7 | Clearly reveal site content. |
| UI-8 | Don't over-format critical content, such as navigation areas. |
| UI-9 | Use meaningful graphics. |
| UI-10 | The site should be easy to navigate and to find what you want. |
| UI-11 | Provide valuable and trustful information. |
| UI-12 | Knowing the website is updated frequently with new information. |
| UI-13 | Customize and target your content/site to your users. Think "one-to-one" websites. |
| UI-14 | Be responsive to the Web user with slow internet connection speed. |
| UI-15 | Be interactive. |
| UI-16 | Have a secure and automated server. |
| UI-17 | Minimize the users' memory load. |
| UI-18 | Give feedback to the user. |
| UI-19 | Prevent code errors. |
| UI-20 | Sitemap required. |
| UI-21 | Maximize accessibility of your contents. |
| UI-22 | The website should attract customers by providing valuable and up-to-date information rather than attractive images or videos. |
| UI-23 | Users should be able to register to become a member for better services. |

Table 19: Requirements of Web-base user interface from good web design

| Requirement ID | Description |
|---|---|
| IE-10 | Be able to classify business activities in news articles into investments, acquisitions, products and marketing, posts, layoffs, contracts and orders, ownership, mergers and only extract events of such types. |
| IE-11 | For security domain, only extract events related to cross-border crime and be able to classify them into migration, human trafficking, smuggle or crisis events. |
| IE-12 | Be able to further classify security events into more detailed sub-types of the four types defined in IE-11. |
| IE-13 | Be able to merge same events extracted from the same article or different articles. |
| IE-14 | Generate XML files with pre-defined slots to store the extraction results and send them to the users as soon as possible after the extraction. |

Table 20: Requirements of PULS IE system elicited from actual users

| Requirement ID | Description |
|---|---|
| DSS-1 | Provide *list view* (Figure 15) that merges same events, groups similar events and filters out same documents coming from different news sources. *Similar events* here means different according to the domain. For example, in disease outbreak domain, similar events are those same disease outbreaks happened in the same location during the same period. |
| DSS-2 | Provide dynamic *timeline view* (Figure 16) that displays events chronologically. |
| DSS-3 | Provide dynamic *graph view* (Figure 18) for business domain which can interactively present relations among business companies, persons, products, etc. |
| DSS-4 | Provide dynamic *map view* (Figure 17) which visualizes the occurrence and frequency of events geographically. |
| DSS-5 | Modify the *table view* (Figure 13) to be more informative and interactive. (E.g., be able to sort by any column ascending or descending; search using Regular Expressions; color-coding different types of events, etc.) |
| DSS-6 | Provide company and sector *profile view* (Figure 19) for business domain. |
| DSS-7 | Allow users to *save their queries* and apply them later. |
| DSS-8 | Provide *edit function* on document view to allow users to correct any slot of an event. |
| DSS-9 | Provide *add new event function* on document view to allow users to add a new event that hasn't been picked up by PULS IE system. |
| DSS-10 | Provide *report problem function* on document view. |
| DSS-11 | Provide *add notes or comments function* on document view. |
| DSS-12 | Provide *restore document function* on document view to allow users to revert any change made to the document by DSS-8, DSS-9, DSS-10 and DSS-11. |
| DSS-13 | Provide links of next and previous events sorted by their time-of-arrival on document view. |
| DSS-14 | Provide links of related events (e.g., the same disease outbreak events in the same country) on document view. |
| DSS-15 | Provide function on document view to export any event extracted by PULS as an XML formated file. |
| DSS-16 | Maintain search constraints among different views. |
| DSS-17 | Provide customization ability for users to set up their prefered views. |
| DSS-18 | Provide view for all events and separated views for different event types in business or cross-border security domain. |

Table 21: Requirements of PULS DS system elicited from actual users

| Requirement ID | Description |
|---|---|
| DB-4 | Be able to store different types of business or security sub-events. |
| DB-5 | Be able to store events added by users which are not found by PULS IE system for all domains. |
| DB-6 | Be able to store additional information of an event coming from the users to accumulate knowledge base for all domains (e.g. user's notes, comments, verified information, etc.). |
| DB-7 | Store the original information of the user-verified event to be able to restore an event to its original state for all domains. |
| DB-8 | Be able to store user's information and different preferences for all domains. |
| DB-9 | Be able to store relations among entities in business domain for supporting graph view (Figure 18). |
| DB-10 | Store geographical data of locations and countries for supporting map view (Figure 17) for all domains. |
| DB-11 | Since grouping events (e.g., same disease in same country within a short period) on the fly might be slow, store grouping information for better supporting list view for all domains. |

Table 22: Requirements of PULS database from specific users' requirements

| Requirement ID | Description |
|---|---|
| IE-15 | Develop a relevance classifier using supervised machine learning algorithms to indicate the relevance of an extracted event in all domains. |
| IE-15-S1 | Define the relevance classification criteria to be used by IE-15 for all domains. |
| IE-15-S2 | Accumulate user rated training/testing data for IE-15 in all domains. |
| IE-16 | Accumulate any valuable knowledge for improving PULS IE system in all domains. |
| IE-16-S1 | For business domain, accumulate sector tags of articles, political persons, descriptors or names of business companies, products or persons, etc. |
| IE-16-S2 | For medical domain, accumulate new infectious diseases or new mutations, etc. |
| IE-16-S3 | For cross-border security domain, accumulate event type, descriptors or names of items, etc. |

Table 23: IE requirements elicited from researchers' point of view

| Requirement ID | Description |
|---|---|
| DSS-19 | Provide rating system on document view and list view to allow users to rate utility/relevance of an event according to relevance classification criteria. |
| DSS-20 | Provide the surveillance view which displays only events with higher relevance (4 and 5) and the complete view. |
| DSS-21 | Provide different display style (color, font-weight, background, etc.) to distinguish low relevance and high relevance events. |
| DSS-22 | Combine all available knowledges PULS has accumulated for a specific company or sector on profile view in business domain. |
| DSS-23 | Provide knowledge view in business domain. |

Table 24: DSS requirements elicited from researchers' point of view

| FR ID | Requirement IDs | FR Description | Domains |
|---|---|---|---|
| FR1 | IE-1,IE-11,IE-12,DB-1,DB-4 | Develop and integrate an IE sub-system for cross-border security domain into PULS IE system for extracting events related to cross-border crime and be able to classify them into migration, human trafficking, smuggle or crisis events, and when possible their sub-types. | security |
| FR2 | IE-2,DB-2 | Develop and integrate a French pipeline into PULS IE system. | medical |
| FR3 | IE-3,DB-2 | Develop and integrate a Russian pipeline into PULS IE system. | security |
| FR4 | IE-4,DB-3 | Integrate the new source of raw news into PULS IE system. | business |
| FR5 | IE-5,DB-2 | Extend the French pipeline and Russian pipeline to other domains. | all |
| FR6 | IE-10,DB-4 | Classify business activities in news articles into investments, acquisitions, new products and marketing, posts, layoffs, contracts and orders, ownership, mergers and only extract events of such types. | business |
| FR7 | IE-13 | Merge same or very similar (according to some similarity metric) events extracted from the same article or different articles. | all |
| FR8 | IE-14 | Generate XML files with pre-defined slots to store the extraction results and send them to the users as soon as possible after the extraction. | all |
| FR9 | IE-15,IE-15-S1,IE-15-S2 | Develop a relevance classifier, using supervised machine learning, to indicate the utility of each extracted event. In order to achieve this, clearly define the relevance classification criteria to be used by IE-15 and accumulate user rated training/testing data for IE-15. | all |
| FR10 | IE-16,IE-16-S1,IE-16-S2,IE-16-S3 | Accumulate additional valuable knowledge along with extracting events for improving PULS IE system. | business |

Table 25: Functional requirements for PULS IE system's users

| FR ID | Requirement IDs | FR Description | Domains |
|---|---|---|---|
| FR11 | DSS-1, DSS-18, DSS-19, DB-11 | Provide interactive *list view* which displays events in groups of related or similar events (based on some similarity metric) and allows users to rate relevance of any event or group. | all |
| FR12 | DSS-2, DSS-18 | Provide dynamic *timeline view* which displays events chronologically. | all |
| FR13 | DSS-3, DB-9 | Provide dynamic *graph view* which interactively presents relations among entities. | business |
| FR14 | DSS-4, DB-10 | Provide dynamic *map view* which visualizes the occurrence and frequency of events geographically. | all |
| FR15 | UI-3,DSS-5, DSS-18 | Improve *table view* to be more informative and interactive. (E.g., be able to sort by any column ascending or descending; search using Regular Expressions; color-coding different types of events, etc.) | all |
| FR16 | DSS-6, DSS-22 | Provide *profile view* for entities. | business |
| FR17 | DSS-7 | Allow users to *save their queries* and apply them later. | all |
| FR18 | UI-4,DSS-8-15, DSS-19, DB-5-7 | Improve *document view* to be more interactive and allow users to make comments to, add, edit, rate or export as xml any event in the document. | all |
| FR19 | DSS-16 | Maintain the user's current search constraints when searching among different views. | all |
| FR20 | DSS-17, DB-8, UI-13 | Provide customization ability for users to set up their prefered views. | all |
| FR21 | DSS-20 | Provide surveillance view which only shows events with high relevance (4 or 5) and complete view which shows all events. | all |
| FR22 | DSS-23 | Provide *knowledge view* for developers to check accumulated knowledges in knowledge base. | business |

Table 26: Functional requirements for PULS DS system's users

| NFR ID | Requirement IDs | NFR Description | Domains |
|---|---|---|---|
| NFR1 | IE-2, IE-3, IE-5, UI-2, DB-2 | Change encoding system to UTF-8 in PULS for handling French and Russian characters. | all |
| NFR2 | IE-6-8 | Increase quality of PULS IE system extraction results. | all |
| NFR3 | IE-9 | Create more suitable test sets for better evaluation purpose. | all |
| NFR4 | UI-5 | Make the website's purpose clear by explaining who you are and what you do. | all |
| NFR5 | UI-6, UI-15, UI18, UI-20 | Help users find what they need and be more interactive. | all |
| NFR6 | UI-7, UI-8, UI-10, UI-21 | Clearly reveal site content, be easy to navigate and maximize accessibility of the contents. | all |
| NFR7 | UI-9, UI-14, UI-17 | Be responsive to the Web user with slow Internet connection speed. Such as using meaningful graphics and minimizing the users' memory load. | all |
| NFR8 | UI-16 | Have a secure and automated server and make sure data stored in the central database must be secure. | all |
| NFR9 | UI-19 | Prevent code errors. | all |
| NFR10 | UI-11, UI-12, UI-22 | The website should attract customers by providing valuable and up-to-date information rather than attractive images or videos. | all |

Table 27: Non-functional requirements for PULS

| Priority\FR ID | FR1 | FR2 | FR3 | FR4 | FR5 | FR6 | FR7 | FR8 |
|---|---|---|---|---|---|---|---|---|
| **High** | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ |
| **Medium** | | | | | | | | |
| **Low** | | | | | ✓ | | | |
| | FR9 | FR10 | FR11 | FR12 | FR13 | FR14 | FR15 | FR16 |
| **High** | ✓ | | ✓ | | ✓ | | ✓ | ✓ |
| **Medium** | | ✓ | | ✓ | | ✓ | | |
| **Low** | | | | | | | | |
| | FR17 | FR18 | FR19 | FR20 | FR21 | FR22 | | |
| **High** | | ✓ | | | | | | |
| **Medium** | ✓ | | ✓ | ✓ | | | | |
| **Low** | | | | | ✓ | ✓ | | |

Table 28: Functional requirements priorities

| Priority\NFR ID | NFR1 | NFR2 | NFR3 | NFR4 | NFR5 | NFR6 | NFR7 |
|---|---|---|---|---|---|---|---|
| **High** | ✓ | ✓ | | | | ✓ | |
| **Medium** | | | ✓ | ✓ | ✓ | | |
| **Low** | | | | | | | ✓ |
| | NFR8 | NFR9 | NFR10 | | | | |
| **High** | | | | | | | |
| **Medium** | | | ✓ | | | | |
| **Low** | ✓ | ✓ | | | | | |

Table 29: Non-functional requirements priorities

# Appendix 3. Implementation details

| **Action** | **Client** | **Server** |
|---|---|---|
| Start up | | Lisp creates initial page with 20 recent records |
| Query/sorting | Javascript sends post request from user query/sorting | Lisp query 20 recent records based on query/sorting and re-create the page |

Table 30: Implementation of table view

| **Action** | **Client** | **Server** |
|---|---|---|
| Start up | | Lisp creates initial page of selected event |
| Update event | Javascript sends post request of update action (modify/make comment/rate relevance/add notes) | Lisp updates the database and re-creates the page |
| Add event | Javascript sends post request of adding event | Lisp creates new event in the database and re-creates the page |
| Toggle edit mode | Javascript sends post request | Lisp re-creates the page |
| Export as xml | Javascript sends post request | Lisp creates the XML file and re-creates the page |

Table 31: Implementation of document view

| Action | Client | Server |
|---|---|---|
| Start up | | Lisp creates initial page of recent 10 groups of events |
| Select view | Javascript sends get request for selected view | Lisp re-creates the page with selected view |
| Toggle group | Javascripts handles it and displays/hides the DOM objects | |
| Trash group | Javascript sends post request and reset the page using AJAX | Lisp updates the database |
| Rate/trash event | Javascript sends post request and resets the page using AJAX | Lisp updates the database |

Table 32: Implementation of list view

| Action | Client | Server |
|---|---|---|
| Start up | Javascript creates the timeline box uses SIMILE [SIM12] which takes the json file as the input | Lisp queries database with selected events and generates a json file |
| Search/highlight | Javascript searches and highlights events in the timeline box | |
| Go to date | Javascripts handles it and resets the timeline box beginning with the selected date | |
| Select event-type | Javascript hides events which are not belonging to the selected event-type | |
| Navigates/check detail | Javascript handles them | |

Table 33: Implementation of timeline view

| Action | Client | Server |
|---|---|---|
| Start up | Javascript creates the map box using Google Maps API [Goo12] which takes the json file as the input | Lisp queries database with selected events and generates a Json file |
| Navigate/check detail | Javascript handles them | |

Table 34: Implementation of map view

| Action | Client | Server |
|---|---|---|
| Start up | Javascript creates the graph using BMVis visualization tool [BMV12] which takes the bmv file as the input | Lisp queries graph core to get selected node and its neighbors and generates a bmv file |
| Search | Javascript searches the nodes and hides un-required nodes in the nodes list | |
| Zoom in/out | Javascript zooms in or out the node in the graph | |
| Hide node's type | Javascript sends post request with node's type to hide | Lisp re-creates the bmv file which does not contain any node with selected type and reset the page |
| Navigate | Javascript handles them | |

Table 35: Implementation of graph view

| **Action** | **Client** | **Server** |
|---|---|---|
| Start up | Javascript convert these objects into designed types of information boxes using Jquery TagCloud [Tag12] and Highcharts [Hig12] according to Section 4.2.4 | Lisp queries database and knowledge base for related information of a business entity specified by user, creates basic HTML DOM and Json objects |
| Compare entities | Javascript sends post request; converts Json object to chart box | Lisp queries database and knowledge base for all entities in comparison and creates the Json object containing the data |
| Navigate | Javascript handles them | |

Table 36: Implementation of profile view

# Appendix 4. Testing examples

| **Testing type** | **Testing details** |
|---|---|
| Input testing | **Generate Template** unit accepts a list of Lisp event objects[a] extracted from a news article as input. Each event contains a number of attributes in the format of Lisp entity object[b]. |
| Output testing | After receiving the input, **Generate Template** can process and generate expected response file output containing all events in designed template format. For example, a disease outbreak event object will be converted to the corresponding disease template object and be printed to a response file. |
| Error handling testing | A number of error handling testings have been performed during and after the implementation to ensure the unit can handle expected errors and would not crash during the process. For instance, given an event object with incorrect type of attribute, the unit is able to report a problem; Given an event object with insufficient attributes, the unit can ignore the missing attributes; Given an event object with unknown type, the unit generates no response without hanging, etc. |

[a] An event object is a Lisp object which represents a fact extracted from the news, e.g., a disease outbreak fact in a country.

[b] A Lisp entity object represents an attribute of an event, e.g., the disease name or country of an outbreak event

Table 37: Unit testing example of PULS IE system

| **Testing type** | **Testing details** |
|---|---|
| Visual testing | **Content displaying**: clear and correct (accuracy of information, grammar, spelling, printable, picture off, colorless testing for color blind people by printing the page in black and white). **Navigation display**: correct and consistent. **Font resizing**: Since all font styles are defined in css file for PULS DS system which uses relative size rather than absolute size, all letters can be resized by selecting the font size in the setting of browsers. |
| Access control testing | Since **table page** is for all users, access is not restricted. For all other pages belonging to member domain and administrator domain, access control testing has been performed. (Testing input: entering the URL directly in the URL address bar of browsers. Testing output: users see the login pop up box, and the content is shown only when users log in and they have the right permission.) |
| Error handling testing | A number of error handling tests have been performed during and after the implementation to ensure the unit can handle expected errors. For example, placing country name into disease search box will produce no result; inputing characters like ", ', [, ], (, ) or direct SQL query text into the search box will not break the search, etc. |
| Interact testing | By inputing search contents into search boxes and clicking return, the corresponding search results have been retrieved and displayed correctly. |

Table 38: Unit testing example for **Table Page** in PULS DS system

| **FR15: improve *table view* to be more informative and interactive.** | |
|---|---|
| **Features** | **Confirm or comments** |
| Provide more advanced query for attribute | Besides only Regular Expression query, table view now supports operators including "=, !=, <, <=, >, >=, not, is null". These operators allow users to search much more accurate information than before. For example, users can compose a query "doc_date > 2011.12, disease not cholera, country = UK". This will give users all cholera outbreak events happened after Dec 2011 only in country UK, not including ones in Ukraine. |
| Provide hierarchical search for location | Users can now search for "Europe" and get all events happened in European countries. |
| Provide sorting by attribute | Yes |
| Apply saved query | Users can select any saved query to view on table view.. |
| Provide special views from legend | Yes |
| Separate views for different event types | Yes for business and security events |
| Distinguish events by color | Events reviewed by users have different colors according to the relevance users have decided. |
| Provide combined view links | Users can query for events they are interested in from table view and then go to the corresponding timeline, map or profile views containing only the searched events. |
| Provide graph view link for business event | Yes |

Table 39: An example of feature checklist

Table 40: An example of usability test

| **Issue** | **Solution** |
|---|---|
| The edit box is not visible on small screen since it is fixed on the page and is not re-sizable. | Use relative position for the edit box; make the edit box re-sizable; move the buttons for accept, fix, report problems on the top of the box. |
| Comments for document made by users are not displayed anywhere else except on document page and hence they are not traceable | Add *notes* field in edit box which allows users to make notes for an event. The notes is then displayed on both table view and list view and it is search-able on table view. |
| Informations for events and document that contains the events are mixed in the edit box | The informations are separated into two boxes. |
| Rerunning a document takes a little bit too long to wait | Change the rerun to only database operation instead of reprocess the documents; add full rerun for extra feature; change function to parse the response file quickly; add loading image while rerunning. |
| There is no way to navigate to other documents in document view | Add next/previous event links; add related information box |
| Related information box makes the whole document page much slower to load. | Use Ajax to load the related information box separately after the document page has been created. |
| Better to have counts for related informations | The counts are added |
| There are too many clicks to assign relevance for an event | Add star rating system for assigning relevance so that users do not need to open edit mode to assign relevance; allow users to select "edit mode always on" in preference settings. |

Continued on Next Page...

Table 40 – Continued

| **Issue** | **Solution** |
|---|---|
| It is good to have auto-complete for some attributes like country, disease for medical, company for business, etc. | Add auto-complete feature for these fields using information extracted from document text. |
| Layout of document view is not tidy | Separate information into different boxes and add a title to each box to indicate its content; choose better background color for reading. |
| Different users have different opinions about the background | Preference setting allows them to select their favorite background. |
| For business events, there is no way to check multiple sectors for current event | Add a pop up window for querying multiple sectors of an event. |